PL/SQL Developer 6.0

用户指南

2004 年 6 月



allroundautomations

目录

1. 介绍

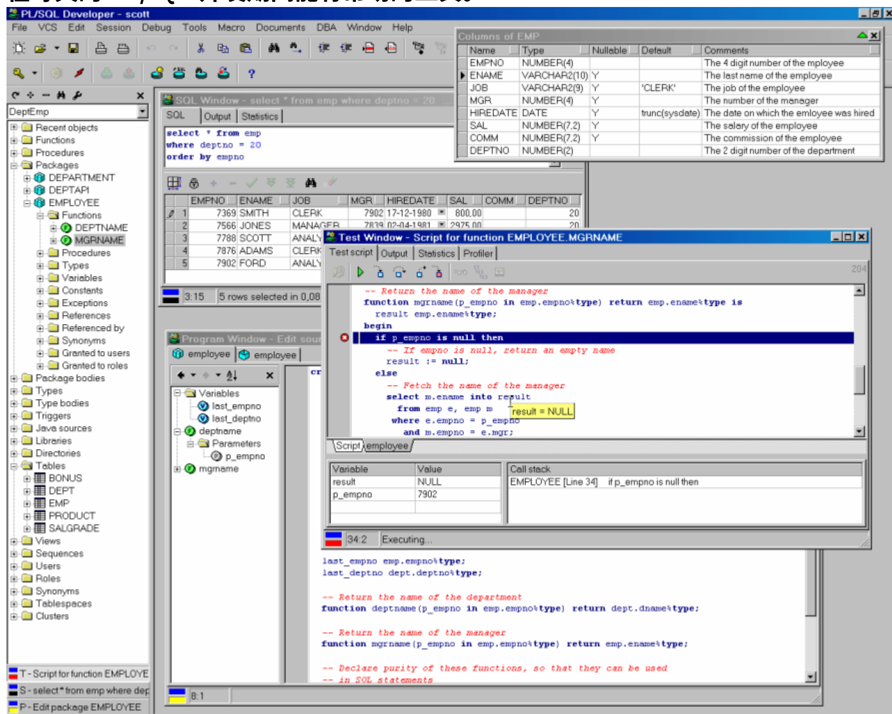
PL/SQL Developer 是一个为 Oracle 数据库开发存储程序单元的集成开发环境(IDE)，使用 PL/SQL Developer 你能方便地创建你的客户/服务器应用程序的服务器部分。

到目前为止从最差处想，你还可以完成如下工作：

- 使用文本编辑器写程序单元（过程、触发器等等）。
- 使用 Oracle SQL*Plus 编译源文件。
- 如果有编译错误，你必须找出它位于源文件何处，纠正它，转回到 SQL*Plus 重新编译它，然后再找下一处错误。
- 使用 SQL*Plus 或你的客户端应用程序测试程序单元。
- 如果发生运行时间错误，你又要很辛苦地定位问题的原因并纠正错误。
- 使用解释计划工具或 tkprof 工具优化你的 SQL 语句。
- 使用 SQL*Plus 或另外的工具在你的数据库里查看或修改其它对象和数据。

这些任务 - 编辑、编译、纠正、测试、调试、优化和查询 - 在不离开 PL/SQL Developer IDE 的情况下都能被完成。此外，PL/SQL Developer 还提供了其它几个

在每天的 PL/SQL 开发期间能有帮助的工具。



编辑

与其它严肃的开发环境一样， PL/SQL Developer 也可以在磁盘上存储你的源文件，其它工具只让你在数据库里编辑源文件，而这不允许进行任何版本控制或部署方案。源文件能通过 SQL*Plus 运行，所以你不使用 PL/SQL Developer 也能在任何平台上部署它们。你可以通过标准的多文档界面马上就能编辑很多文件。

编辑器为程序员提供了很广的协助范围。对于 SQL 语句和 PL/SQL 语句，它有语句前后关系的帮助，我们都已经提供在那儿了：你开始键入 *substr*， 但已经忘了参数的准确含义，现在你可以简单地单击 *F1* 就能获得 *SQL 参考手册*的相应主题。同样地，在编辑器里面卷起的窗口里，表、视图和程序单元都能为你提供描述。代码助手被集成在编辑器里面，当你键入数据库对象的名称时，它会自动地显示出数据库对象的信息，允许你从这些信息里浏览和挑选选项。对于大的包或类型体，程序编辑器为每一个导航都提供了带有代码内容的树形视图，提供了前后浏览按钮，提供了超连接导航。查询设计器允许你图形化地创建 select 语句。 PL/SQL Developer 的可扩展模板使你往你的程序里插入标准的 SQL 和 PL/SQL 代码变得很容易。所有编辑器都使用恰当的 SQL，PL/SQL 和 SQL*Plus 的语法高亮使你的代码更具有可读性。

编译与纠错

在编辑器里，你可以编译源文件。如果发生编译错误，你会自动地被带到相应的源代码行，所有编译错误都被记录在编辑器底部的一个列表里。

由于你可以编译源文件而不用保存它，你就能安全地探索解决问题的很多可选方法。编辑器保留着你已经修改了文件而没有保存或编译它的踪迹。

测试与调试

要测试你的程序单元，你可以写一个程序脚本，这个测试脚本的主体包含了你可以规划测试代码的 PL/SQL 块，任何你要在 PL/SQL 块里使用的变量都能被声明、被赋输入值，还能在运行后被查看。

当你运行一个测试脚本，也许会出现运行时间错误。在这种情况下， PL/SQL Developer 允许你查看错误堆栈的源，每一个涉及运行时间错误的错误堆栈源行都会被高亮显示，所以你能很容易地返回到有问题的地方。

如果你正在使用 7.3.4 或更高版本的 Oracle，你就可以使用 PL/SQL Developer 的集成调试器。你可以一步一步地通过代码、设置断点、查看/设置变量、查看调用堆栈等等。

在 Oracle8i 以及更高版本上，你还可以使用 *dbms_trace* 特性把程序运行的已选事件记入日志，这能帮助你分析程序流和异常。

来自于调用到 *dbms_output* 和 *PL/SQL Web 工具箱*的输出会自动地在相应的测试窗口标签页里被显示出来。

优化

要在你的程序单元里优化 SQL 语句， Oracle 的解释计划工具能是一个好帮手，所以它集成在 PL/SQL Developer 的 IDE 里面。在源文件里简单地选择 SQL 语句并按 F5，查询计划就会真实地出现在一个单独的解释计划窗口里，然后你就能在源文件之外通过修改语句来优化查询计划，再把它复制回去。

对于已运行的 SQL 语句和 PL/SQL 程序单元，你还能查看统计表，这些统计表能包括耗时、CPU 时间、逻辑读、物理读、物理写等等的内容。

Oracle8i 引进了 PL/SQL 概览图，它允许你概览你的 PL/SQL 代码。对于每个已运行的 PL/SQL 代码行，你能测定运行时间以及已经运行了多少次。

查询

要查询数据库里的数据，你可以使用 SQL 窗口运行任何 SQL 语句。所有已运行的语句都被保留在历史缓冲区里，这样你能容易地重新运行它们。任何查询结果都会被方便地显示在一个单独的表格里，随后你可以在这个表格里插入、更新或删除记录。结果表格还能被用于查询实例模式，以便你能容易地查找你需要的信息。

要查询数据库对象，你可以使用对象浏览器。数据库对象的所有相关属性，例如表、视图、序列、函数、过程、包、类型、触发器等，包括对象之间的依赖关系，都能被

显示出来。浏览器使用了树形视图，这类似于容易地使用以指向并点击方式浏览的 Windows 的资源管理器。

运行 SQL 脚本

PL/SQL Developer 包括了一个能被用于运行 SQL 脚本或运行 SQL 语句的命令窗口，你还可以运行你可能熟悉的与 SQL*Plus 命令很相似的命令。

创建与修改表定义

你可以容易地创建与修改带有使用任何 SQL 语句的表定义，正好填入到对话窗口里的定义里面，以修改列、约束、索引、权限、存储信息、注释等等。你可以在数据库里应用这些定义，并且查看、修改和保存因而产生的 SQL 。

报告

PL/SQL Developer 伴生了许多基于 HTML 格式的标准报告。你可以在 PL/SQL Developer 里面查看这些报告，可以打印它们或者另存它们为 HTML 文件，你还可以创建你自己定制的报告，报告能从报告菜单里简便易行地制成。

工程

要组织你的工作，你可以使用 PL/SQL Developer 的工程概念，工程由很多文件和数据库对象组成，这些对象通过工程项目窗口很容易被建成，并且通过单击鼠标就能被编译。

工具

PL/SQL Developer 提供了几个能对开发有帮助的工具，这些工具包括允许你在数据库对象源里搜索文本的查找数据库对象工具、快速编译在开发期间已经变得无效的对象的编译无效对象工具、导出和导入工具、创建测试数据的数据生成器工具、导出用户对象 DDL 语句的导出用户对象工具、比较两个用户对象定义的比较用户对象工具、会话信息工具、事件监视器。

除了这些标准工具之外，你还可以定义你自己的工具，并把他们包括在 PL/SQL Developer 的工具菜单里面。

2. 安装

主要有两种方法安装 PL/SQL Developer：

- 工作站安装：在这种情况下你要在每个被使用的工作站本地安装这个软件。
- 服务器安装：在这种情况下你要在服务器上从每个被使用的工作站能访问到的地方安装这个软件。

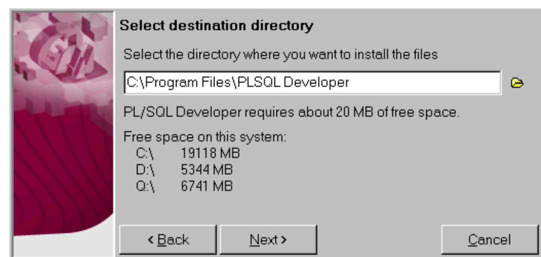
两种安装方法在这里都将被解释。

2.1 系统需求

PL/SQL Developer 是 32 位的 Windows 应用程序，因此你的工作站上必须至少安装了 Windows 95 或 Windows NT 4。要连接到 Oracle 数据库，PL/SQL Developer 需要 32 位版本的 SQL*Net、 Net 8、 Net 9 或 Net 10。

2.2 工作站安装

要在本地工作站上安装 PL/SQL Developer，运行来自于安装媒体的 setup 程序，然后下面的对话框将会显示出来：



你可以为程序文件选择一个目的目录、一个开始菜单的文件夹、一个 PL/SQL Developer 在桌面上的快捷方式的位置以及安装选项，然后按页末的完成按钮，这样程序文件就会被复制，快捷方式也会被创建。

2.3 基于服务器安装

对于多用户许可，你可以在文件服务器上创建一个单一的 PL/SQL Developer 安装，并且对所有许可用户都提供访问权限。做到了这一点，与前一节描述的方法一样，你就可以运行服务器上的 setup 程序了。你必须选择使用这个软件的用户的共享目的目录。当 PL/SQL Developer 第一次在工作站上运行时，这种安装方式会被自动地检测到，然后不用用户干预本地安装就会完成。

PL/SQL Developer 能被安装到用户只读访问的目录里，不过有点例外情况：

- 宏子目录必须可写，因为所有的宏都要储存到这里。

- 首选项子目录必须可写，因为个人首选项集将被储存到这里。请看第 14.31 章的详细资料。

2.4 脚本安装

除了上面描述的安装过程之外，你还可以创建无人值守安装的脚本，请阅读包含在 *install.txt* 里面的详细信息。

2.5 卸载 PL/SQL Developer

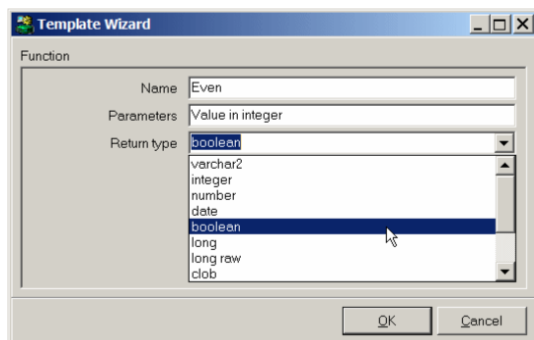
若因为某些原因你希望卸载 PL/SQL Developer，你可以使用控制面版里的添加/移除程序来移除它。

3. 编写程序

在 Oracle 数据库里面，你能识别出五种不同类型的存储程序单元：函数、过程、包、类型和触发器，程序编辑器允许你用同一个方法创建和修改这五种类型的程序单元。

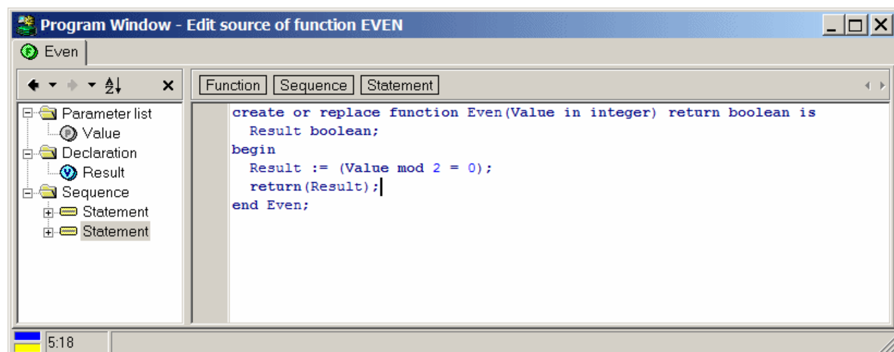
3.1 创建程序

要创建一个新程序，请按工具栏里的*新建*按钮，选择*程序窗口*项目，例如*函数*子项目，你现在会得到与这个程序类型相关的不同变量的提示，这样我们就知道在哪里创建函数、函数名、参数列表以及它的返回值类型：



与这个对话框的标题所暗示的一样，这个信息来自于模板。PL/SQL Developer 提供了几个标准的能按照你的需要修改的模板，你也可以定义新的模板，关于定义模板的信息被提供在第 20 章里。

你输入完了变量并按了*确定*按钮以后，程序编辑器窗口出现了，里面有模板函数。每个你在程序编辑器里创建的程序单元都一定以常见的 SQL 语法 'create or replace' 开始，这样我们就创建了一个函数 'even'，源文件如此图所见：

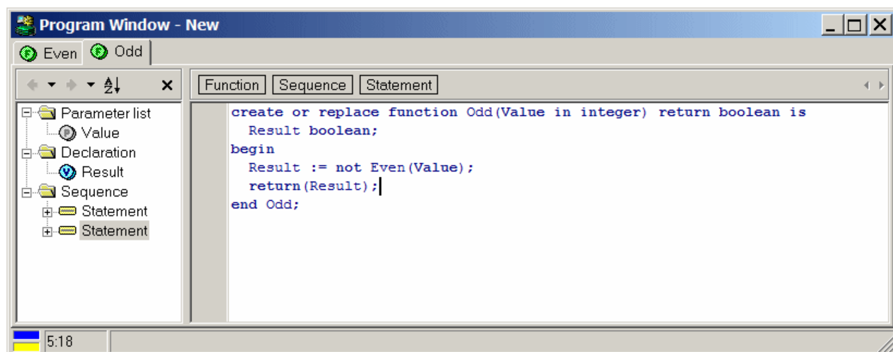


在编辑器的左边，你可以看到显示程序单元结构的代码目录，这被用于诸如包体和类型体等这样大的程序单元的导航，这个特性在第 16.11 章里有详细的描述。

在编辑器的上方，你可以看到显示来自于光标位置的层次的代码层次，这样你可以定位在*函数*里语句的*序列*里的*语句*位置（返回语句）。对于复杂程序，准确地看到你定位于何处是很有用的，这能帮助你选择 PL/SQL 代码的某一部分。详细的描述在第16.12 章里。

程序文件能包含不止一个程序单元。在程序编辑器里右击鼠标，一个允许你添加或删除程序单元的弹出式菜单就显示出来了。你可以通过选择窗口上方适当的跳格键在程序单元之间互相切换，这个方法能使你方便地保持相关的程序单元都在一个源文件里，包说明与体就是这个特性的一个很好的例子。

程序单元应该放在可能与之相参照的程序文件里的其它任何程序单元的后面。如果你创建了一个与前面创建的函数 'even' 相参照的函数 'odd'，则程序编辑器应该是这样的：



3.2 保存程序

你可以通过按工具栏里的*保存*按钮来保存程序文件，一个显示一些程序文件标准扩展名的保存对话框就会显示出来了：

程序类型	扩展名
函数	.fnc
过程	.prc
包说明与体	.pck
包说明	.spc
包体	.bdy
类型说明与体	.typ
类型说明	.tps
类型体	.tpb
触发器	.trg

Java 源	.jsp

你可以通过如第 14.26 章所描述的首选项来改变这些扩展名。如果只有一个单一的函数、过程、包、类型、触发器或 java 源在程序文件里，则文件名会自动地从这个程序单元的名称和类型得来。文件保存以后，文件名就在窗口标题里显示出来。

已保存的程序文件有一个与 Oracle SQL*Plus 完全一致的格式，包含 'odd' 与 'even' 函数的源文件格式如下：

```
create or replace  function Even(Value in integer) return boolean is
  Result boolean;
begin
  Result := (Value mod 2 = 0);
  return(Result);
end Even;
/
create or replace function Odd(Value in integer) return boolean is
  Result boolean;
begin
  Result := not Even(Value);
  return(Result);
end Odd;
/
```

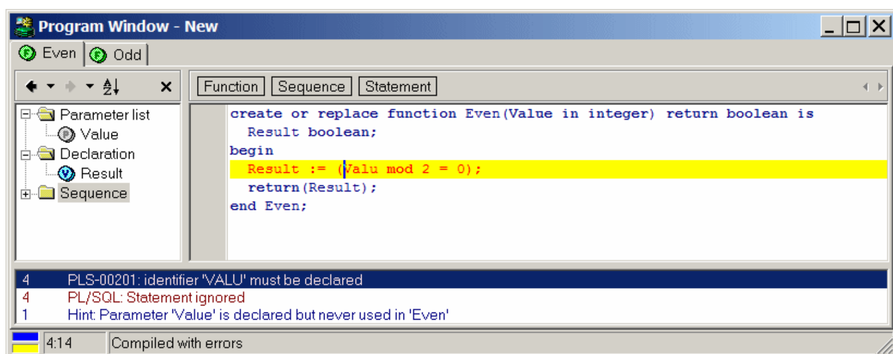
当这个文件通过 SQL*Plus 被运行的时候，两个函数都会被创建。

3.3 修改程序

通过按工具栏上的*打开*按钮并且选择*程序文件*项目，你可以打开以前保存的程序文件。如果文件在最近用过，通过选择*文件*菜单上的*重新打开*或者点击工具栏中打开按钮旁边的箭头，你也可以打开它。当你开始编辑时，窗口底部有一个蓝色的指示器就亮起来了，这意味着文件已经被修改，但还没有被保存。还有一个黄色指示器，它意味着文件已经被修改，但还没有被编译。

3.4 编译程序

通过按工具栏中*执行*按钮，你可以编译程序文件。程序文件里的所有程序单元都会被编译，从第一个开始编译，不管你当前选了哪个程序单元。当错误出现时，编译被终止，编辑器被定位在引起错误的源行：



如果有不只一个错误并且你希望纠正它们，你点击程序编辑器底部的下一个错误信息就可以转到有错误的地方。如果你已经为 PL/SQL Developer 配置了如第 23.2 章里所描述的 HTML 手册，你在编译错误上双击就可以显示出 *Oracle 错误信息*手册里的相关段落。

编译错误列表也能包含提示。提示的内容不是引起编译失败的原因，但它们可能显示出能引起运行时间错误的问题（例如与 null 比较、函数没有返回值、无用参数等等）。关于提示的更多信息，请看第 14.17 章。

如果在编译后有错误信息显示在信息框里，这说明创建语句失败，事实上源文件没有被编译，错误信息应该能解释错误的原因。例如，如果你得到的信息是 'create or replace' 语法错误，信息框里会显示"ORA-00900：无效的 SQL 语句"。

注：如果你使用的 Oracle 服务器是 7.2 或更早版本，触发器编译错误总是被报告在信息框里。Oracle 服务器自从 7.3 版本以后，触发器编译错误采用与过程、函数、包、类型一样的方式被报告。

4. 测试程序

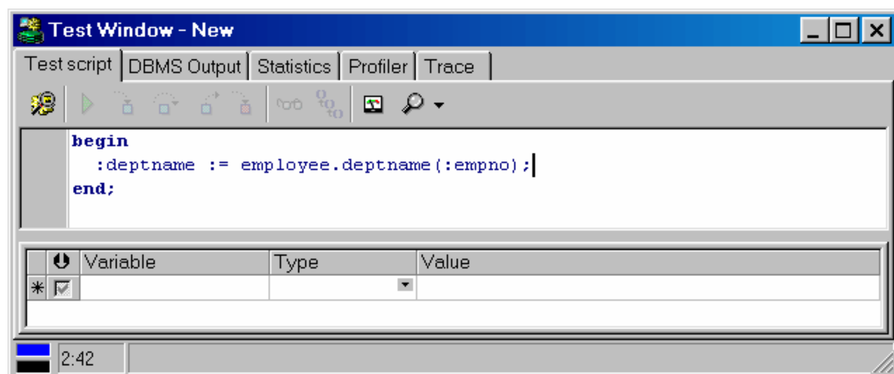
成功地编译了程序以后，你还需要测试它。要达到这一目的，你可以使用 PL/SQL Developer 的测试脚本。测试脚本允许你运行一个或多个程序单元，允许你定义输入、输出和输入/输出变量，允许你查看变量和给变量赋值。如果在测试脚本的运行期间出现运行时间错误，你可以查看涉及错误的程序单元的源。

如果你使用的是 Oracle 7.3.4 或以后版本，你可以使用 PL/SQL Developer 的集成调试器。你能一步一步地运行你的代码，设置断点，查看和设置变量，查看调用堆栈，等等。

4.1 创建测试脚本

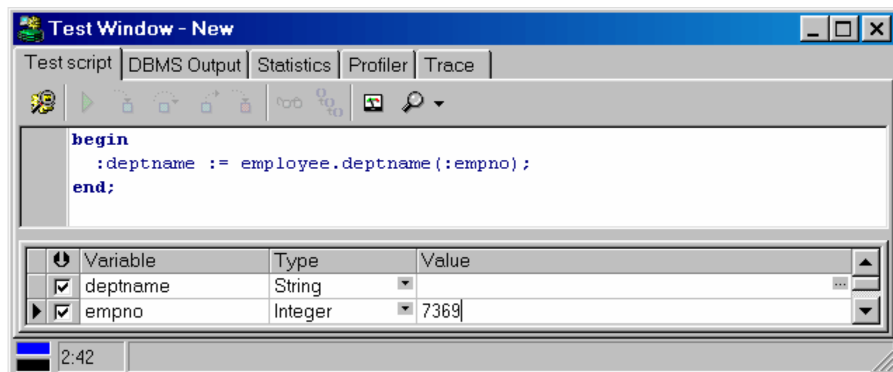
有两种方法创建测试脚本。你可以简单地右击浏览器里的一个程序单元（一个独立的函数或过程、一个打包的函数或过程、或者一个对象类型方法），选择*测试*项目。这将创建一个新的测试脚本，这个测试脚本带有参数变量和结果变量。为了说明问题，我们将从头开始用手工创建一个测试脚本。

要创建空的测试脚本，请按工具栏里的*新建*按钮，选择*测试窗口*项目。你现在可以用类似于 declare...begin...end 这样的语法在测试脚本的正文内键入一个 PL/SQL 块。作为一个例子，我们将测试包 *employee*，运行一些雇员/部门演示表的雇员函数。函数 *deptname* 返回雇员部门的名字，并且能使用下列 PL/SQL 块来测试：



这个 PL/SQL 块包含了一个简单的我们要测试的函数的调用，它使用了变量 *deptname* 和 *empno* 。我们必须给 *empno* 赋一个值，然后运行，检查 *deptname* 的值以确定函数是否返回了正确的值。

用冒号作为这些变量的前缀，这些变量能被声明于测试脚本窗口的底部：

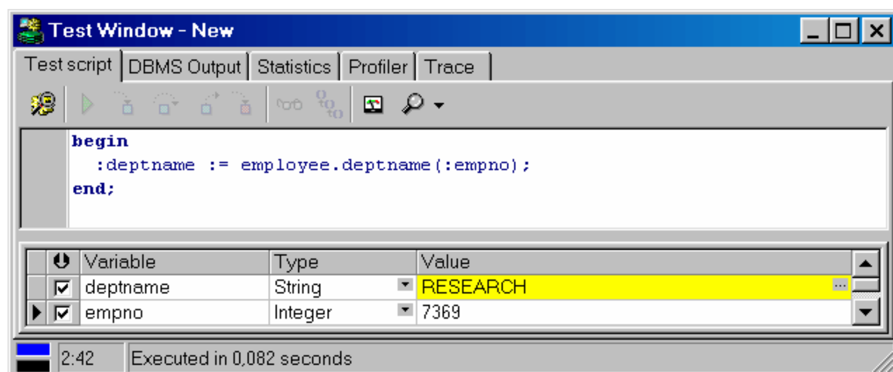


变量列表左上方的***扫描变量源***按钮（⏼）能被用于快速复制源里的变量。在此之后，整型数据类型和值 7369 （来自于调查部的 Smith 先生）就被赋给了 *empno* 。现在，我们已经做好运行脚本的准备了。

你也可以在 PL/SQL 块本地的声明部分来声明变量，但是你将不能修改或者查看这些变量的值。这些本地变量的优势是你能使用记录类型、 PL/SQL 表、类型等等。

4.2 运行测试脚本

要运行测试脚本，请按工具栏里的***执行***按钮。在运行完后，变量的值就被更新了，任何改变了值的变量都会用黄色的背景被显示出来：



函数运行的结果是 'RESEARCH' （调查部的调查。——译者姜华东注），显然它的运行是正确的。在窗口底部，运行时间被显示出来了，这个信息能被用于优化你的运行代码。为了优化目的，你也可以通过选择统计表标签页来查看 PL/SQL 块运行情况的统计表。通过在运行脚本前按***创建概览图报告***按钮，你还可以创建每个 PL/SQL 代码已运行行的概览报告。在运行完后，你就可以转到***概览图***页来查看报告。统计表和概览图分别在第 5.2 章和第 5.3 章里有详细解释。

你可以通过按*中断*按钮来终止一个正在运行的测试脚本，这在程序陷入死循环或者运行时间长于预期的时候显得特别有用。要注意的是按*中断*按钮并不总是成功的。例如，如果程序正在等待一个锁，它并不响应中断信号。

运行了一个脚本后，一个事务也许就已经被已运行的程序单元开始了。若是这样，工具栏里的*提交*和*回滚*按钮将被激活。要了解关于事务的更多信息，请看第 12 章。

4.3 变量类型

测试窗口变量部分的类型列表包含了能被使用的所有类型，这些类型的特性将在这一章里被解释。

类型	描述
Integer	能被用于数字值，范围为 -2^32 到 2^32 - 1。
Float	能被用于其它数字值。
String	相当于表里的 varchar2 数据类型，能容纳 2000 (Oracle7) 或 4000 (Oracle8) 个字符。此数据类型的值的单元格里有一个能调用文本编辑器的单元格按钮，所以你能容易地查看和修改多行字符串。
Date	日期和时间的数据类型。
Long	Long 数据类型不在表格里显示它的值，只显示 <Long>，按单元格按钮可以调出文本编辑器。
Long Raw	按 Long Raw 值的单元格按钮将调出一个导入/导出文件对话框，允许你载入或保存这个变量的内容。
Cursor	此数据类型用于 PL/SQL 里你能使用游标变量的地方。运行了测试脚本后，你可以按值的单元格按钮以在 SQL 窗口里显示结果集。
CLOB，BLOB 与 BFile	在你能查看 LOB 数据以前，这些 LOB 定位变量必须在服务器上被初始化。CLOB 将调用文本编辑器， BLOB 与 BFile 将调用导入/导出文件对话框。
PL/SQL String	为 PL/SQL 使用此数据类型， varchar2 值可以达到 32000 个字符。
Char	定长字符串（空格被填补）。
Substitution	没有绑定变量的限制，置换变量就能被使用。在它被发送到服务器以前，PL/SQL 块里的变量名会被它在文本里的值置换。
临时 CLOB 与 BLOB	在它们通过测试脚本被传送到服务器以前，这些 LOB 定位器已经可以持有数据了。

Boolean 变量

当你选择变量类型列表框，你会注意到缺少 Boolean 数据类型，这是因为 SQL*Net 不支持这种数据类型。要使用 Boolean 变量，你可以声明它为整型，并且使用 Oracle 提供的 *sys.diutil.bool_to_int* 与 *sys.diutil.int_to_bool* 函数。这些函数能在 null/true/false 与 null/0/1 两者之间进行转换。如果你在浏览器里使用测试函数，这个转换会为你自动地产生。

4.4 保存测试脚本

要保存测试脚本，请按保存工具栏里的*保存*按钮。保存对话框使用 *.tst* 为测试脚本的默认扩展名，尽管你可以使用第 14.26 章里描述的首选项来改变默认扩展名。PL/SQL 块、所有变量（名称、类型与值）以及调试监视器（请看第 4.10 章）都被保存下来。由于保存了测试脚本，你以后若要修改它，就能容易地重新测试程序单元了。

通过按工具栏里的*打开*按钮并选择*测试脚本*，你能打开以前保存的测试脚本，这将创建一个新的测试窗口。作为一种选择，你可以右击先前创建的测试窗口，选择*载入*项目，这将在现有的测试窗口里打开测试脚本。

保存为 SQL*Plus 脚本

作为一种选择，你可以用 SQL*Plus 兼容格式保存测试脚本，这允许你在不能用 PL/SQL Developer 的环境里运行脚本（例如在 Unix 服务器上）。要做到这一点，请在文件选择器对话框里选择 *SQL*Plus 脚本 (*.sql)* 作为文件类型。来自前一章的例子脚本就将被这样保存：

```
rem PL/SQL Developer Test Script

set feedback off
set autoprint off

rem Declare variables
variable result varchar2(2000)
variable p_empno number

rem Set variables
begin
  :result := null;
  :p_empno := 7369;
end;
/

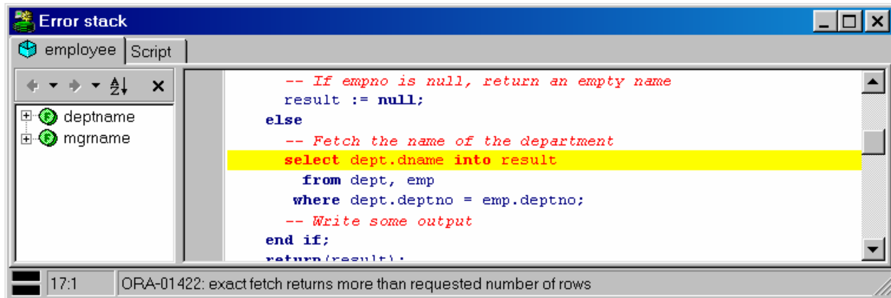
rem Execute PL/SQL Block
begin
  :result := employee.deptname(:p_empno);
end;
/

rem Print variables
print result
print p_empno
```

注意：你不能再把 SQL*Plus 脚本作为测试脚本打开。如果你要重新使用测试窗口，你还是把它保存为测试脚本为好！

4.5 跟踪运行时间错误

无论何时你的程序单元引起了运行时间错误， PL/SQL Developer 都将问你是否希望查看错误堆栈的源。这个方法能使你快速查找错误的原因。如果在我们的 *deptname* 函数里有一个错误，则错误堆栈窗口将显示如下：



在顶部你能看到每个与运行时间错误相关的程序单元的标签，程序单元按从右往左的顺序被调用。你可以通过标签来浏览，能容易地找到导致错误的程序流。在这里它显示出测试脚本已经调用了函数 *employee.deptname* 。这显然能获得更多复杂的、甚至通过触发器的信息。

注：如果你使用的是 Oracle 服务器 7.2 或者更早版本，触发器源将不被显示在错误堆栈窗口里。

错误堆栈高亮显示每一与错误相关的红色的行。对于堆栈里的最近的程序单元，这是一个引起错误的行。对于所有其它的程序单元，这是一个调用下一个程序单元完成的行。

4.6 包声明和 Java 会话声明

当你正在编辑、编译、测试包时，包声明在每一个编译后将被 Oracle 抛弃，所有包里的全局变量都将被复位，并且初始化块将被再运行一次。这可能导致意外的结果。无论 PL/SQL Developer 何时检测到这种情况，在测试窗口的状态行都会有一个警告显示出来：“警告：ORA-04068：现有的包声明已经被抛弃”。

同样地，如果你编辑和编译了 Java 源，任何现有的 Java 会话都会被清除。PL/SQL Developer 也将处理这种情况，将在状态行上显示：“警告：ORA-29549：类<类名>已经被改变，Java 会话声明被清除了”。

4.7　查看结果集

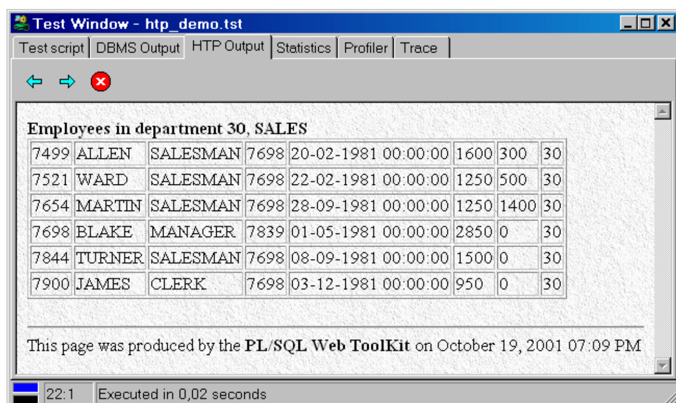
在测试脚本里，你并不会仅被限制于 PL/SQL 块。你还可以运行单个 SQL 语句，包括 select 语句。运行 select 语句将显示在另外的*结果*标签页里，这里可以显示所有的已选行。

4.8　查看 dbms_output

为了调试目的，“打印”某些信息到来自于程序单元内部的屏幕上可能是必要的。要达到这个目的，Oracle 创建了 *dbms_output* 包。通过调用 *dbms_output.put_line*，信息就被放进了输出缓冲区。在运行了测试脚本后，你可以选择测试窗口顶部的输出标签来查看输出缓冲区的内容。在此页上，你还可以设置输出缓冲区的大小或者允许/禁止缓冲。输出页的默认设置被如第 14.4 章里所述的输出首选项所控制。

4.9　查看 HTP 输出

如果你要利用 PL/SQL Web 工具箱来测试程序单元，则 HTP Output 标签页将被显示出来（如果有 HTP 输出的话）：



如果没有 HTP 输出，这个标签页将是不可见的。

4.10　调试

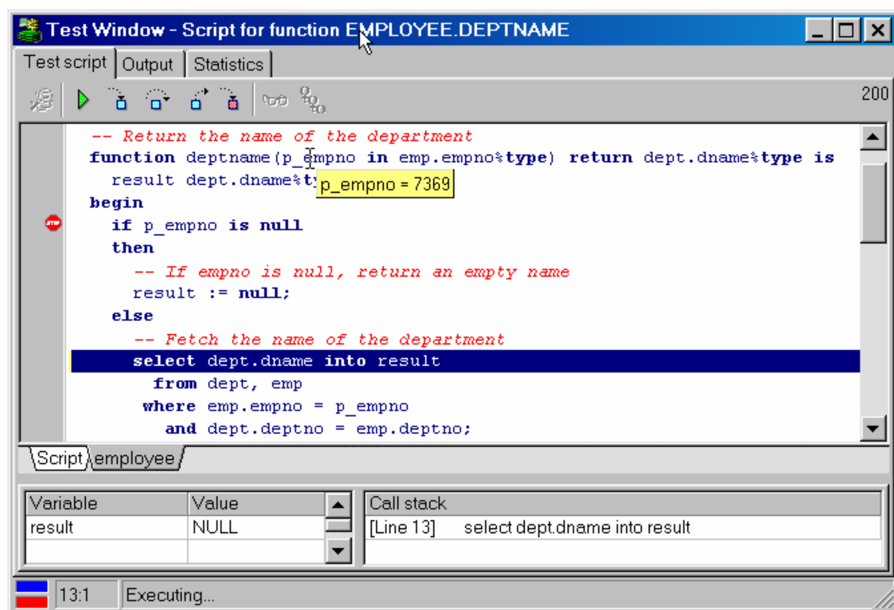
对于那些设计错误，跟踪起来确实很辛苦，好在测试窗口提供了一个集成调试器。在窗口的顶部，你会发现一个与调试器有相关功能的工具栏，要开始调试会话，只需要按调试工具栏左边的*开始*按钮就可以，而不是按主工具栏里的执行按钮。现在，其它按钮都被激活了，你可以准备调试了。

控制运行

调试器开始运行后，运行将暂停于测试脚本的第一个语句之前，你可以使用调试工具栏里的按钮来控制运行：

- 运行脚本直到完成。
- 单步进入过程、函数或调用下一行的方法。如果下一行包含了能引起触发器工作的更新、插入、或删除语句，你将单步进入这个触发器。
- 单步跳过下一行。它将被运行，但你将不会单步进入到源。
- 单步退出当前的程序单元。
- 运行到异常出现。运行将被暂停在引起异常的行。在下一步之后，事实上异常将被解除。

无论你何时单步进入程序单元，它的源将自动地被载入到测试窗口。现在，编辑器面板的底部将显示出每个程序单元的标签，因此你能容易地在它们之间互相切换，以查看源、设置/移除断点等等。通过右击编辑器面板，你能从测试窗口移除程序单元（如果你不再对它有兴趣的话）：



查看与设置变量值

要在调试会话期间查看变量值，你可以简单地移动鼠标的指针越过源里的变量。半秒以后，它的值将被弹出显示。测试脚本的 PL/SQL 块里的变量可以从来都不被显示。复杂的变量（例如记录与对象类型）也不能被显示，尽管你可以查看单独的字段。未来版本的 Oracle 也许会解除这些限制。

要设置变量值，在程序源里右击它的名字，在弹出式菜单里选择*设置变量*项目，一个带有变量当前值的输入区就显示出来了，你可以输入一个新值并按*回车*来应用它，或者按 *ESC* 取消操作：



在同样的弹出式菜单里，你还可以选择添加变量到监视列表，这个列表意味着在每个调试步骤以后，变量值将被自动地显示和更新在测试窗口左面底部的监视列表里。

如果变量是一个标量数据类型的集合（PL/SQL 表、数组或嵌套表），你就可以通过右击它并选择来自于弹出式菜单的*查看集合变量*来查看整个集合。

要注意如果程序单元被编译带有调试信息，变量值仅能被查看与设置。这是首选项的存在造成的，它会使每一个编译都带有调试信息。你可以采用手工方式添加调试信息：右击浏览器里的程序单元，选择来自于弹出式菜单上的*添加调试信息*项目。

使用断点

断点能被用于暂停程序在你的 PL/SQL 代码的某一行上的运行。当运行暂停时，你可以查看和设置变量、一步一步地通过代码等等。你可以定义断点的条件，当这个条件被满足时，运行将被暂停。对于每一个断点，你可以定义一个断点到达时放在输出页的信息。

设置断点

有两种方法设置断点：在程序窗口或在测试窗口。对于这两种方法，你只需要简单地点击编辑器左边适当的行的空白处就可以。断点标记将显示出来，这标志着断点已经存在于那个行上了。当你在调试模式里运行测试窗口，如果有一个断点被遇到，运行将停止。

如果你在程序窗口里设置断点，这可能是这个特殊的程序单元还仍然没有被编译到数据库里，因此这种断点现在还仍然不能被应用到数据库里。在这种情况下，断点的标志将有不同的显示。当你在后来编译程序单元的时候，断点将被应用，相应的标志将改变到实际情况。只要程序单元不被编译，任何先前在这个程序单元里设置的断点都会是有效的。

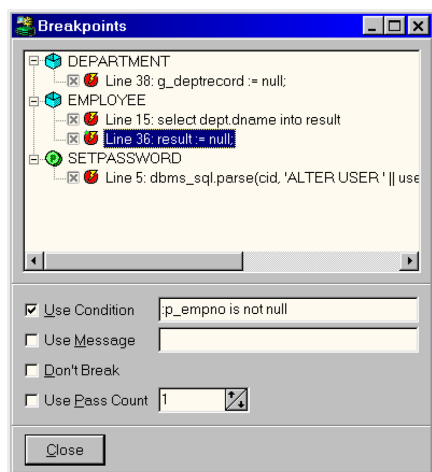
你不能在测试脚本的 PL/SQL 块里设置断点。

要删除断点，再一次点击断点标志就可以了。

断点条件

有时候即使你仅仅对某一环境下程序状态有兴趣，你还是需要在经常被运行的行上定义断点。在这种情况下，你可以为断点定义一个条件，运行仅仅在条件满足的时候才能暂停。

要为断点定义条件，请右击断点标志，选择来自于弹出式菜单的*修改断点*项目，下列对话框就被显示出来：



在这个对话框里，你可以看到所有有断点的程序单元，断点被列在它们下面。每一个断点都有一个能被用于允许或禁止它的检验栏，底部区域显示如下字段：

- **使用条件** – 这个检验栏允许或禁止使用条件，条件自己应该是布尔表达式。当运行到达断点行时，只有条件值为 True 运行才被暂停。在条件里，你可以使用任何 SQL 表达式，你还可以使用任何在断点位置已知的变量。在交互调试期间，这些是你能查看或设置的相同的变量。变量前面应该加冒号。例如，如果在断点位置 *ename* 是个有效的变量，则 upper(:ename) = 'SMITH' 是一个有效的条件。
- **使用信息** – 这个检验栏允许或禁止使用信息。当运行到达断点行时，如果条件得到满足，信息将被放到输出页上。
- **不中断** – 这个检验栏只在你还要定义一个信息时有用。当它为已检验时，运行从不暂停在断点行上。 用此方法你可以定义仅仅在输出页上生成信息的断点。
- **使用通过计数** – 这个检验栏允许或禁止通过计数，这个通过计数定义断点行在运行被暂停前必须被通过多少次。例如，如果你定义通过计数为 10，则运行在断点行每到达 10 次就暂停一次。

查看调用堆栈

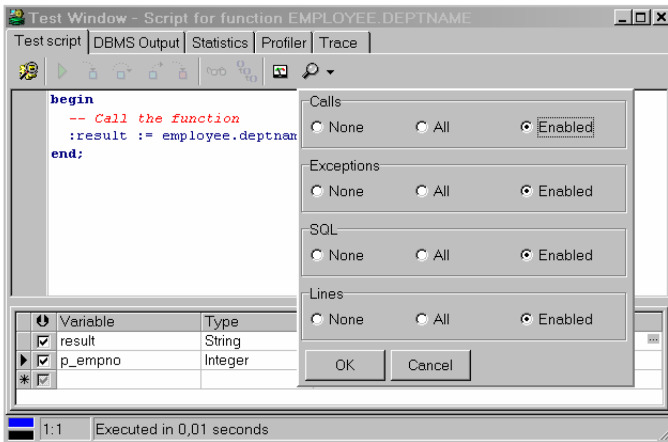
调用堆栈能被查看于测试窗口的右侧底部，它在每个调试步骤到达后能够自动更新。

调试器首选项

在工具菜单的首选项项目里，你会发现有与调试器相关的首选项。这些首选项被详细描述在第 14.3 章里。

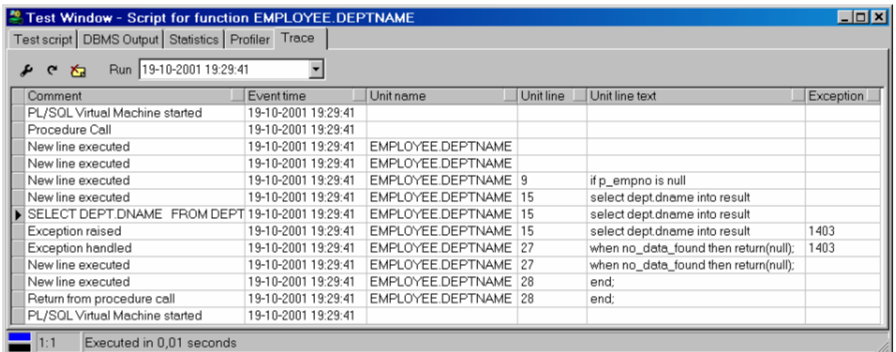
4.11 跟踪运行

如果你正在使用 Oracle8i 或更新版本，你可以使用测试窗口的跟踪工具去跟踪你的 PL/SQL 代码的运行情况。你可以设定你要跟踪的事件：调用、异常、SQL 或者甚至每一个 PL/SQL 代码的可运行行。按*选择跟踪级别*按钮，下一步按*创建跟踪报告*按钮，就可以看到下列配置屏幕：



你可以控制你是否要跟踪每一个程序单元的特定时间（*所有*）、仅仅在那些与调试信息一起编译的程序单元（*允许*）或无（*无*）。

要创建跟踪报告，简单地按测试窗口工具栏里的创建跟踪报告按钮并且运行你的测试脚本就可以。在运行完后，你可以转到跟踪标签页去查看报告，并且查看先前运行的报告：



Test Window - Script for function EMPLOYEE.DEPTNAME

Test script | DBMS Output | Statistics | Profiler | Trace

Run 19-10-2001 19:29:41

Comment	Event time	Unit name	Unit line	Unit line text	Exception
PL/SQL Virtual Machine started	19-10-2001 19:29:41				
Procedure Call	19-10-2001 19:29:41				
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME			
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME			
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	9	if p_empno is null	
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	15	select dept.dname into result	
SELECT DEPT.DNAME FROM DEPT	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	15	select dept.dname into result	
Exception raised	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	15	select dept.dname into result	1403
Exception handled	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	27	when no_data_found then return(null);	1403
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	27	when no_data_found then return(null);	
New line executed	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	28	end;	
Return from procedure call	19-10-2001 19:29:41	EMPLOYEE.DEPTNAME	28	end;	
PL/SQL Virtual Machine started	19-10-2001 19:29:41				

1:1 Executed in 0,01 seconds

在这个（简单的）报告里，你可以看到函数 *employee.deptname* 的第 15 行的 select 语句出现了异常 1403 （没有找到数据），这个异常在同一程序单元的第 27 行被处理了。

对于每个跟踪时间，你可以包含一些信息在报告里，就像事件描述、时间、程序单元、行数、异常等等。要配置这个，请按跟踪工具栏里左上方的*首选项*按钮，这将调来如第 14.5 章里所描述的跟踪首选项页。

要查看旧的跟踪报告，就从*运行*选择列表选择先前的跟踪运行。按*删除运行*按钮可以删除当前的已选的运行。

要在 Oracle 跟踪工具里获得更多信息，请看 *Oracle 供应包参考*手册里的 *DBMS_TRACE* 。

5. 优化

要优化你程序单元里的 SQL 语句，Oracle 的解释计划工具能对显示语句的运行路径有很大的帮助，因此它被集成在 PL/SQL Developer 的 IDE 里。

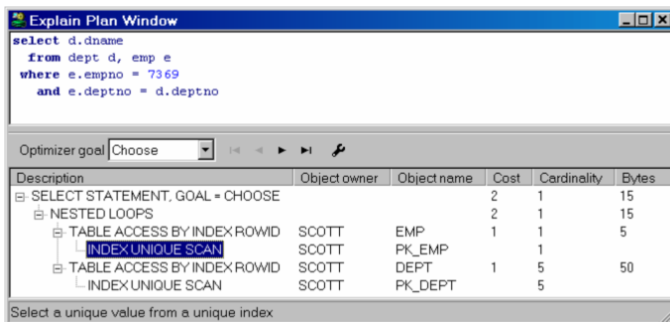
要查看 SQL 语句或 PL/SQL 程序单元实际源的使用情况， PL/SQL Developer 能够显示关于它运行情况的统计表。你可以配置你希望显示的统计表，并且能包括耗时、CPU 时间、逻辑读、物理读、物理写等等。

要确定每一个单独的 PL/SQL 代码行的运行时间，你可以使用 PL/SQL 概览图。这个特性对于 Oracle 8.0 或更早版本是不可用的。

最后，通过激活 SQL 跟踪，你可以使用 Oracle 的 *tkprof* 工具来获得程序单元里关于所有已运行的 SQL 语句的资源的使用信息。

5.1 使用解释计划窗口

要使用 Oracle 的解释计划工具，请按工具栏里的*新建*按钮，选择*解释计划窗口*。此时一个空的解释计划窗口就显示出来了。在窗口的上半部，你可以键入你希望分析的 SQL 语句。在按了工具栏里的执行按钮后，运行计划就被显示在窗口的下半部：



你现在可以改变 SQL 语句，再按执行按钮看看这个改变的效果。要了解关于解释计划的更多信息，你可以阅读 *Oracle 服务器调整*手册。

要看查询计划里各种优化器目标的效果，请选择*优化器目标*列表里相应的条目，这个计划将立即被更新到反映这些改变上。

使用*第一个*、*下一个*、*上一个*和*最近操作*按钮操纵查询计划的操作顺序。计划已经被确定后，第一个操作将高亮显示。

解释计划工具使用所谓的计划表来储存运行计划。如果这个表对当前用户不可用，PL/SQL Developer 将问你是否要在当前用户的计划里创建这个表。要定义哪几个列来自于你要看的计划表，是什么顺序，请按*首选项*按钮，这将调出首选项页（请看第 14.20 章）。

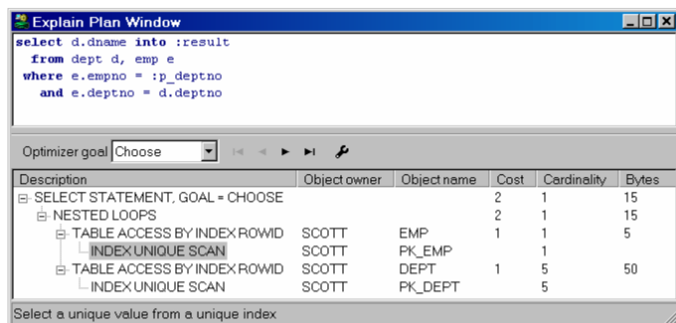
注：如果你用的是 Oracle 服务器 7.2 或更早版本，耗费与基数在计划表里不可用。如果你用的是 Oracle 服务器 7.3 或更高版本而耗费与基数不被显示，你大概需要更新计划表。

如果你右击*对象名列*，这将为已选对象调出弹出式菜单。

在程序单元里分析 SQL 语句

大多数时间里，你希望分析的 SQL 语句都被包含在程序单元的源里。要做这个，你可以选择程序编辑器里的 SQL 语句并且选择来自于*帮助*菜单的*解释计划*（或按 F5），一个新的解释计划窗口将为已选的 SQL 语句创建。

当来自于程序单元的 SQL 语句包含了 PL/SQL 变量并且你运行解释计划窗口时，你会收到一个"ORA-00904：列名无效"的错误，解释计划工具错误地把这些变量都当成了列，因此每一个变量都必须加上一个冒号前缀。如果我们已经要求分析 *employee.deptname* 函数里的 SQL 语句，则 result 与 *p_empno* 变量肯定会按如下被改变：



现在解释计划工具知道了哪一个标识符是列、哪个标识符是 PL/SQL 变量。

5.2 自动统计

当你在 SQL 窗口或者测试窗口执行语句时， PL/SQL Developer 将自动地生成一个此执行的统计报告。有一个条件是你必须在动态性能表 *v$session*, 、*v$statname* 与 *v$sesstat* 里有 select 权限（这些表通过标准的 *plustrace* 角色提供）。

你可以通过改变到 SQL 窗口或测试窗口顶部的统计表标签来查看统计报告。用于 *employee.deptname* 函数运行的统计表可能显示如下：

Test Window - New

Test script | DBMS Output | Statistics | Profiler | Trace

Name	Last	Total
CPU used by this session	0	0
physical reads	5	12
physical writes	0	0
session logical reads	17	45
sorts (disk)	0	0
sorts (memory)	2	2
sorts (rows)	5	5
table fetch by rowid	1	6
table scan blocks gotten	3	5
table scan rows gotten	19	21
table scans (long tables)	0	0
table scans (short tables)	3	5

Executed in 0,084 seconds

对于每一个统计值，你可以看到最近运行的值和当前会话的合计。你可以通过设置首选项来配置统计表是否被显示或者哪一个值被显示，如第 14.2 章里所描述的那样。统计表的默认设置与你的 SQL 和 PL/SQL 有关，如这儿所描述：

统计值	含义
CPU used by this	CPU 的使用（以百分之一秒为单位计）
Physical reads	从磁盘读出的块数
Physical writes	写到磁盘的块数
session logical	从块缓冲区或磁盘里读出的块数
sorts (disk)	在临时段里被运行的排序数量
sorts (memory)	在内存里被运行的排序数量
sorts (rows)	被排序的行数
table fetch by rowid	按 rowid 获取的行数，通常为索引访问的结果
table scan blocks	为全表扫描读的块数
table scan rows	为全表扫描读的行数
table scans (long	在长表里全表扫描的数量
table scans (short	在短表里全表扫描的数量

你能包含的其它统计表依赖于 Oracle 服务器的版本，不在本手册里被描述。如果你希望获得关于它们的信息，有很多从事于这个主题的 Oracle 调整的书可用，*Oracle 服务器参考手册*也简要地描述了这些统计表。

统计表能被导出到 CSV 文件（逗号分隔值），以便以后能用电子表格软件打开它。只要右击统计表、选择*导出*项目、选定 *CSV 文件*项目就可以了。二者择一地，你可以选择 TSV 、XML 或 HTML 格式，或者选择复制它到剪贴板。

5.3 PL/SQL 概览图

PL/SQL 概览图是一个帮助你优化 PL/SQL 代码的很强大的工具。对于每一个已运行的代码行，合计时间、最多时间、最少时间、平均时间和运行次数都将被确定。

在测试窗口里，概览图是容易访问的。在运行测试脚本前，简单地按测试窗口工具栏里的*创建概览图报告*按钮就可以了。如果你随后运行脚本，你可以转到*概览图*页查看运行报告。

下面的实例报告显示了在 *employee.deptname* 函数里 select 语句用时 149 毫秒，3 个 dbms_output 调用用时 57 毫秒：

Test Window - deptname.tst

Test script | Output | Statistics | Profiler

Run deptname.tst - 14-01-2001 14:46:18 Unit (All) - 244

Unit	Line	Total time	Occurrences	Text
ANONYMOUS BLOCK	4	27	2	:result := employee.deptname(p_empno => :p_empno);
EMPLOYEE	9	1	1	if p_empno is null
EMPLOYEE	12	0	0	result := null;
EMPLOYEE	15	149	1	select dept.dname into result
EMPLOYEE	20	57	3	dbms_output.put_line(to_char(sysdate, 'dd-mm-yyyy hh24:mi:ss')
EMPLOYEE	24	5	1	return(result);
EMPLOYEE	27	0	0	when no_data_found then return(null);
EMPLOYEE	28	4	1	end;
EMPLOYEE	34	0	0	if p_empno is null then
EMPLOYEE	36	0	0	result := null;
EMPLOYEE	39	0	0	select m.ename into result
EMPLOYEE	44	0	0	dbms_output.put_line(to_char(sysdate, 'dd-mm-yyyy hh24:mi:ss')
EMPLOYEE	48	0	0	return(result);
EMPLOYEE	50	0	0	when no_data_found then return(null);
EMPLOYEE	51	0	0	end;

1:1 Executed in 0 seconds

在默认情况下，*概览图*页将显示最近运行的报告。你也可以从*运行*列表里选择以前的运行。*单元*列表允许你放大一个运行的特定的程序单元。

*合计时间*列显示了一个与行相关的时间的图形化描述，这个图形是与最多时间来做比较的，这允许你快速确定最费时间的行。报告可以通过按列标题里的排序按钮来排序。

如果源行显示为红色，这意味着自概览图被创建以后程序单元已经被改变，因此现在代码被显示的行跟概览图被创建的时候比是不一样的。

通过按*首选项*按钮，你可以把概览图报告改变成不同的版面外观，这将调出相应的首选项页，如第 14.6 章里所述。

关于 PL/SQL 概览图的更多信息，请看在“ *Oracle8i 应用包参考* ”手册里的 *dbms_profiler* 。

注：不是所有平台都能提供精确的时间信息。

5.4 SQL 跟踪

你可以通过按工具栏里的 *SQL 跟踪*按钮来激活 SQL 跟踪，此后所有的由 SQL 窗口或测试窗口而来的服务器进程都将被登录到数据库服务器上的跟踪文件里。有一个前提条件是，你正在使用的数据库实例的定时统计的参数必须被设成 True 。

你可以通过再一次按 *SQL 跟踪*按钮来禁止 SQL 跟踪。

你可以通过使用数据库服务器上的 Oracle *tkprof* 工具来查看跟踪文件里的信息，它将为每一个 SQL 语句产生一个关于耗时、CPU 时间、I/O 等等的报告，因此它能

使你知道程序单元里哪一个语句最耗时间。要学习更多关于 Oracle *tkprof* 的知识，你可以读一读*“Oracle 7 服务器调整”*手册。

运行 *employee.deptname* 函数的 *tkprof* 报告可以显示如下：

```
TKPROF: Release 7.2.2.3.1 - Production on Fri Sep 26 14:59:08 1997

Copyright (c) Oracle Corporation 1979, 1994.  All rights reserved.

Trace file: ora07087.trc
Sort options: default

********************************************************************************
count    = number of times OCI procedure was executed
cpu      = cpu time in seconds executing
elapsed  = elapsed time in seconds executing
disk     = number of physical reads of buffers from disk
query    = number of buffers gotten for consistent read
current  = number of buffers gotten in current mode (usually for update)
rows     = number of rows processed by the fetch or execute call
********************************************************************************

begin
  :deptname := employee.deptname(:empno);
end;

call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.01       0.01          0          0          0           1
Fetch        0      0.00       0.00          0          0          0           0
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        2      0.01       0.01          0          0          0           1

Misses in library cache during parse: 0
Optimizer hint: CHOOSE
Parsing user id: 16
********************************************************************************

SELECT DEPT.DNAME
FROM
 DEPT,EMP  WHERE EMP.EMPNO = :b1  AND DEPT.DEPTNO = EMP.DEPTNO


call     count       cpu    elapsed       disk      query    current        rows
------- ------  -------- ---------- ---------- ---------- ----------  ----------
Parse        1      0.00       0.00          0          0          0           0
Execute      1      0.00       0.00          0          0          0           5
Fetch        1      0.06       0.06          4          4          0           1
------- ------  -------- ---------- ---------- ---------- ----------  ----------
total        3      0.06       0.06          4          4          0           6

Misses in library cache during parse: 0
Optimizer hint: CHOOSE
Parsing user id: 16     (recursive depth: 1)
********************************************************************************
```

6. 专用 SQL

在程序开发期间，你经常需要运行一些 SQL 语句。既要测试语句，又要查看或者更新表里的数据。你可以在 PL/SQL Developer 内部通过使用 SQL 窗口就能做好这件事。

6.1 使用 SQL 窗口

要创建一个新的 SQL 窗口，请按工具栏里的*新建*按钮，再选择 *SQL 窗口*，一个空的 SQL 窗口就显示出来了。在这个窗口里，你可以键入 SQL 语句。如果它是 select 语句，窗口会分成两部分，下半部分是显示结果的表格。如果我们要查看雇员表里的所有雇员， SQL 窗口将显示如下：

SQL Window - select * from emp order by ename

SQL | Output | Statistics

```
select * from emp
order by ename
```

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7876	ADAMS	CLERK	7788	12-01-1983	1100,00		20
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00	300,00	30
3	7782	CLARK	MANAGER	7839	09-06-1981	2450,00		10
4	7902	FORD	ANALYST	7566	03-12-1981	3000,00		20
5	7900	JAMES	CLERK	7698	03-12-1981	950,00		30
6	7566	JONES	MANAGER	7839	02-04-1981	2975,00		20
7	7839	KING	PRESIDENT		17-11-1981	5000,00		10
8	7654	MARTIN	SALESMAN	7698	28-09-1981	1250,00	1400,00	30
9	7934	MILLER	CLERK	7782	23-01-1982	1300,00		10
10	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00		20

2:15 10 rows selected in 0,03 seconds (more...)

SQL 语句被运行， 10 行数据被提取。若结果集大于 10 行，可以通过按窗口与表格之间的*获取下一页*和*获取最后页*来显示结果集，并且*（更多...）*被显示在状态行上。最初只有 10 行被获取，这是因为能被显示在表格里的记录数只有这些。按*获取下一页*按钮将获取下一个 10 行数据，等等。按*获取最后页*按钮将获取所有行。

如果你选择了 SQL 编辑器里文本的一部分，那么仅仅被选择的文本将被运行。这样，你可以在编辑器里有多于一个的语句，一个一个地运行它们。

如果你运行 SQL 语句要花很长时间才能完成，你可以按*中断*按钮来终止它。注意按*中断*按钮并不总是成功的。例如，如果语句正在等待锁，它将不响应中断信号。如果你在 SQL 窗口正在获取行的时候按了*中断*按钮，它将简单地停止并显示已经被获取的行，你现在可以用*获取下一页*和*获取最后页*来继续获取记录。

注意，SQL 窗口首选项允许你确定一个 select 语句最初可以获取多少行，这在第 14.18 章里有描述。

在窗口的右边你可以看到两个按钮，它们会允许你操纵已经输入进 SQL 窗口里的所有 SQL 语句，这样你可以快速地再运行预先输入的语句。

为了优化目的，你可以通过选择*统计表*标签来查看 SQL 语句运行情况的统计表。统计表在第 5.2 章里有解释。

当你打印 SQL 窗口时， SQL 语句和结果表格都将被打印。通过选择结果表格里的行，你可以限制将要打印的行的数量。

6.2 结果表格处理

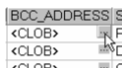
SQL 窗口的结果表格可以用不同的方法来处理。一些单元格类型有特殊的行为，行、列和单元格排列都能被选择和打印，列可以被移动，行可以被排序，你可以转到单一的记录来查看，等等。

识别 null 值

Null 值被显示于一个浅黄色背景的单元格里，所以你可以快速地从全是空格的值里区别出它们。这对识别 null 值和在单元格里不直接显示的值也是有用的，如 long 值和 LOB 值。你可以通过首选项来改变 null 值单元格的颜色（请看第 14.18 章）。

查看大数据列

Long 、 Long Raw 、 CLOB 、 BLOB 与 BFILE 列的值不直接在结果表格里显示，表格里只是简单地显示为 <Long> 、 <Long Raw> 、 <CLOB> 、 <BLOB> 和 <BFILE>：



当你点击某一列的单元格按钮时，大数据编辑器就被调出来了，它允许你查看或编辑不同格式的数据。更多细节请看第 17 章。

你也可以点击字符列上的单元格按钮来调用大数据编辑器，不过单元格按钮仅存在于大于 20 个字符的列。

查看日期列

日期列有一个显示日历的单元格按钮，这个日历的当前日期为高亮显示。如果日期还有时间部分，你可以在日历上这样查看它的值：



查看时间戳列

时间戳列被显示在指定的格式里，这个格式在你的 Oracle 主目录的 NLS_TIMESTAMP_FORMAT 和 NLS_TIMESTAMP_TZ_FORMAT 的注册表设置里被指定。

查看 XML 数据

在数据库里储存 XML 数据有多种情形，都可以储存为： CLOB（Oracle8i）、XMLTYPE（Oracle9i）或者甚至 Varchar2 、 Long 列。只要 XML 格式是以标准的 XML 标题开始，大数据编辑器无论如何都能识别出它并且转入 XML 格式。要了解更多的详细资料，请看第 17 章。

查看嵌套游标列

如果你在 select 语句的字段列表里包含了嵌套游标，最初列值将被显示为 <Cursor>。按单元格按钮将调出新的 SQL 窗口，它带有一个有游标结果集的结果表格，这可以用于查看简单的嵌套查询。注意每一个嵌套游标都将在暗中造成一个打开的游标，因此对于大的结果集，你能容易地陷入 OPEN_CURSORS 的限制，并且得到 *“ORA-01000：打开游标超越最大限制”*的错误。

选择行、列与单元格

要在结果表格里选择行或列，只需点行标题或列标题并且拖动鼠标指针直到要选择的区域高亮显示即可：

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
5	7654	MARTIN	SALESMAN	7698	28-09-1981	1250,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
5	7654	MARTIN	SALESMAN	7698	28-09-1981	1250,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

现在，选择的列可以通过释放鼠标按钮来移动了。再点已选列的标题中的一个，就可以拖动已选列到新的地方了。

要选择一个特定范围的单元格，移动鼠标指针越过一个单元格的左边直到光标形态改变，再按鼠标按钮，拖动鼠标直到要选择的区域高亮显示：

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
5	7654	MARTIN	SALESMAN	7698	28-09-1981	1250,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

要选择所有的行与列，请按左上角的单元格，或者右击表格并选择来自于弹出式菜单的***全选***项目。

已选择的高亮区域通常都能被复制或打印。

导出数据

有几个方法可以导出结果集表格里的数据。运行了 select 语句后，你可以采用如上面所述的方法选择一定范围的单元格，右击它，在从弹出式菜单上选择***导出结果***项目。这将显示出一个子菜单，在这里你可以选定导出数据到 CSV 格式（逗号分隔值）、TSV 格式（制表符分隔值）、HTML 格式或者 XML 格式。选择了格式后，你就能指定导出文件了。

作为选择，通过按 ***Ctrl-C***，或者通过右击你选择的数据并且选择来自于弹出式菜单上的***复制***或***带标题复制***项目，你就可以把你选择的数据复制到剪贴板上。随后，你可以把这些数据粘贴到另外的应用程序里，例如电子表格、文字处理软件等等。

要在 Microsoft Excel 里快速地操作这些结果集信息，就选择***复制到 Excel*** 项目，这将打开一个新的 Excel 实例，并且所有已选择的数据也将被复制过去。

给行排序

要在结果表格里给行排序，请按你要排序的列的标题按钮：

	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL
8	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00
1	7369	SMITH	CLERK	7902	17-12-1980	800,00
11	7876	ADAMS	CLERK	7788	12-01-1983	1100,00
4	7566	JONES	MANAGER	7839	02-04-1981	2975,00
6	7698	BLAKE	MANAGER	7839	01-05-1981	2850,00
7	7782	CLARK	MANAGER	7839	09-06-1981	2450,00
9	7839	KING	PRESIDENT		17-11-1981	5000,00
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00
3	7521	WARD	SALESMAN	7698	22-02-1981	1250,00
10	7844	TURNER	SALESMAN	7698	08-09-1981	1500,00

按一次标题按钮，行被排成升序；再按一次标题按钮，行被排成降序；按第三次，行将撤消排序。按另一个列的标题按钮将对这个列所在的行进行排序，但也将使用先前的排序列为第二排序列（标题按钮上显示为一个圆点）。在上面的例子里，列 ***job*** 是第一排序列，列 ***hiredate*** 为第二排序列。

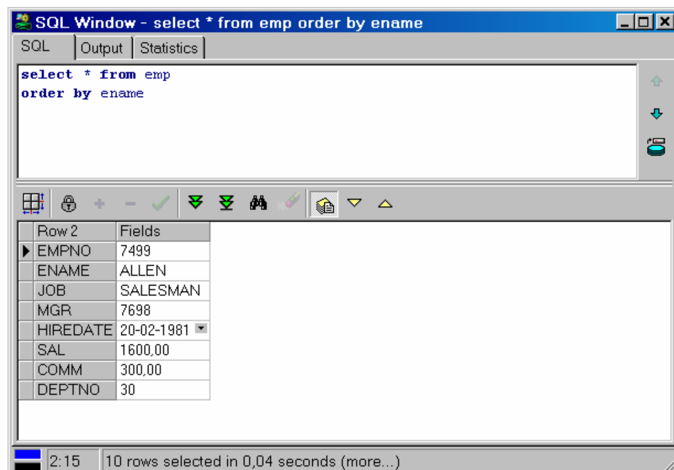
注意排序是在本地被运行的，并且仅仅是已经获取的行的排序。如果你在对结果排序后又获取了另外一些行，这些新行将被添加到结果表格的末尾，并不排序。对于大的

结果集，本地排序能花很长时间。在这种情况下，最好能在 select 语句里使用 order by 子句，让 Oracle 服务器来排序。

单个记录查看

如果结果集包含很多列，每条记录都显示在一行上也许是不方便的，你必须前后滚读相关的列（尽管你可以移动列），不能马上看到记录的所有列。

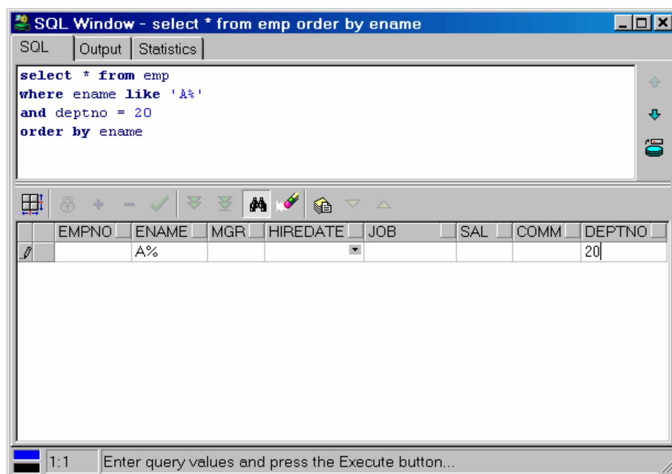
通过按表格工具栏里的*单个记录查看*按钮，你就能一次查看一个单个记录了：



现在每行显示一个列名和值。表格工具栏里的*下一个记录*和*上一个记录*能被用于操纵结果集。要转回到多记录查看，就再按一次*单个记录查看*按钮。

6.3 实例模式查询

运行了 select 语句后，如果你要搜索特定的记录，你可以按结果集上方的*实例查询*按钮。按这个按钮将清除表格，只剩下一个空记录，在这里你可以输入你的查询值。例如，如果你要查找部门编码为 20 的、以字母 A 打头的所有的雇员，就在列 *ename* 里输入 A% 并且在 *deptno* 列里输入 20 即可：



你一输入查询值，原始的 SQL 文本就会被修改以反映出新的查询要求。再按*实例查询*按钮，或按*执行*按钮，将运行已修改的查询，并显示被限定的结果。再按*实例查询*按钮，前面的查询值仍然被保留。按*清除记录*按钮就可以清除查询值了。

查询值不限制带有通配符的单一值，你可以使用下列表达式（也可以选择方括号里的表达式）：

- 值 [= 值]
- 带有通配符的值 [like 带有通配符的值]
- > 值
- != 值 [<> 值]
- in (值 1, 值 2, ...)
- between 值 1 and 值 2
- null [is null]
- not null [is not null]

注意，如果你使用了文字值，你可以省略字符值的引号（例如，在 SQL 文本里，SMITH 将自动地被转换成 'SMITH'）。对于所有其它表达式，你必须提供 Oracle 服务器懂得的文字值（例如，!='SMITH'）。对于数字和日期值，也是这种情形。如果你要在表达式里使用它，就必须使用 Oracle 服务器懂得的格式。

有几个首选项允许你控制实例查询功能的默认行为（例如，区分大小写），第 14.18 章描述了这几个首选项。

6.4 连接查询

当查看一个结果集时，你经常要查询相关表。例如，当你查看 *dept* 表时，你也许要查询一个部门里的所有雇员，或者反之亦然：当你查看 *emp* 表时，你也许要查询一个雇员所在的部门。

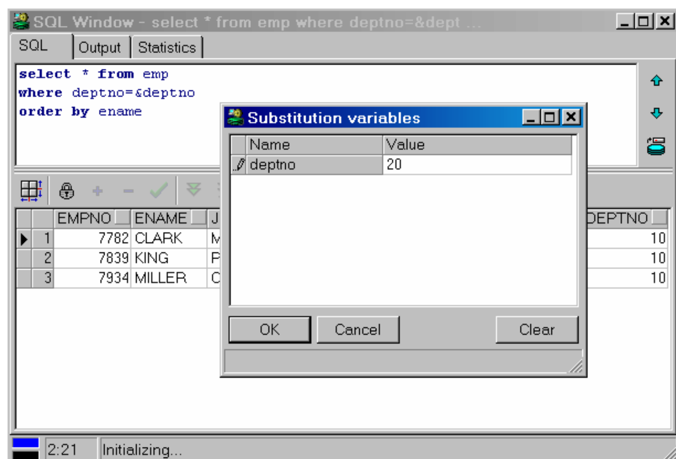
多数情况下，这些父表与子表之间有外键约束，这时 SQL 窗口能够自动地为你生成和运行这些查询。如果你确定了结果集里一条特定记录的位置，按表格工具栏里的*连接查询*按钮，这将显示出一个弹出式菜单，里面包含了当前 select 语句的表的所有父表和子表。例如，如果你在查询表 *emp* ，则下面的弹出式菜单将显示出来：

上面部分显示出了父表（和外键名），下面部分显示了子表。项目 *dept* 将为这个雇员的部门（30）生成一个查询，第一个 *emp* 项目（父表里的）将为这个雇员的经理（7839）生成一个查询，第二个 *emp* 项目（子表里的）将为被这个雇员管理的所有雇员生成一个查询。

连接查询将在同样的 SQL 窗口里被运行，除非你使 SQL 窗口首选项*在新窗口里执行连接查询*（参看第 14.18 章）。当你按住*连接查询*按钮同时再按 *Ctrl* 键时，这个首选项将反向发生。

6.5 置换变量

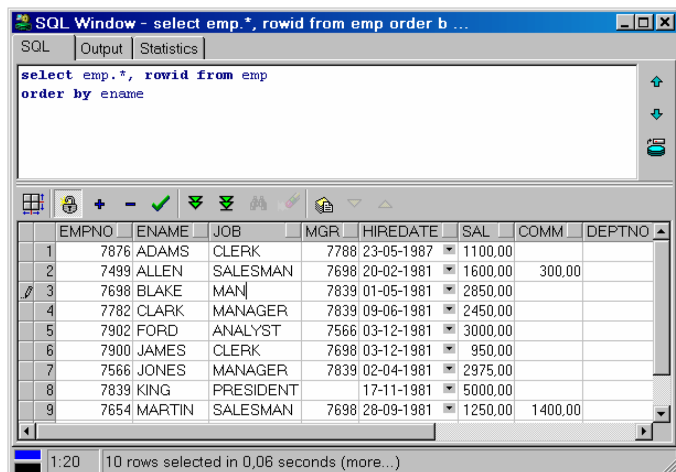
当查询被运行时，你可以在你的 SQL 文本里使用置换变量以允许用户输入数据。置换变量的最简单形式类似于你也许习惯使用的 SQL*Plus ：



通过指定 SQL 文本里的 *&deptno* ，你将被提示为这个变量输入一个值。在运行前，SQL 文本里的变量参照将被置换成这个输入值。你可以指定不止一个的置换变量，而且你可以不止一次地指定一个单一的置换变量。此外，你还可以定义数据类型、默认值、选择列表、列表查询、检验栏等其它高级选项。这些特性也被用于报告窗口，详细情况在第 9 章里有描述。

6.6 更新数据库

要更新、插入或删除数据库里的记录，你可以在 SQL 窗口里运行适当的 DML 语句。通过将 rowid 包括在选择列表里或者使用 *select ... for update* 语句 ，使结果表格可以更新，这大概对我们是更加方便的：



你应该意识到，一个 *select ... for update* 语句将锁住所有已选的记录，所以一般来说包括 rowid 可能是一种使结果集可更新的最好的方法。如果 select 语句是一个内部联接，则第一个表的列就能被更新，所有的其它列都是只读，有别名的列也不能被更新。

如果结果表格是可更新的，你可以按结果表格右边的*编辑数据*按钮来编辑记录，你可以通过按*插入记录*或*删除记录*按钮来插入或删除表格里的记录。如果你已经选择了多条记录，所有被选择的记录都会被删除。

实际上修改结果表格不会改变数据库里的任何东西。要记入已经更新、插入或删除的记录到数据库里，请按*记入改变*按钮。如果你禁止了*自动提交 SQL 窗口*的选项，则如果事务已经被开始，工具栏里的*提交*和*回滚*按钮就会被激活。要了解更多关于事务的信息，请看第 12 章。

编辑大数据列

与前面一章解释的一样，你可以点击 Long 、 Long Raw 、 CLOB 、 BLOB 和 BFILE 列的单元格按钮来查看各种格式的值。当结果表格可更新时，你可以使用大数据编辑器来改变列的值。

6.7 查看和编辑 XMLTYPE 列

SYS.XMLTYPE 类型不被 Oracle Net 9.0 和更早版本支持，所以你将不能直接访问被存储在某一列里的 XML 数据。例如，如果 *xml_text* 是 SYS.XMLTYPE 列，则下列查询将不提供预期的结果：

```
select id, xml_text from xml_table
```

仅仅 *id* 列将在结果集里显示出来。要查看 XML 数据，就要使用 *getclobval()* 成员函数：

```
select id, t.xml_text.getclobval() from xml_table t
```

这样 CLOB 就能在文本编辑器里被看到， XML 语法高亮将自动地被应用。要编辑 XML 数据，就要使结果集可更新，这通常是通过包括 rowid 实现的：

```
select id, t.xml_text.getclobval(), rowid from xml_table t
```

现在你能编辑 CLOB 并且能记入已修改的数据到数据库了。注意，表和视图的*查看数据*与*编辑数据*函数将自动地应用到这些规则里，因此查看或编辑 XMLTYPE 列的最简单方法是右击表或视图，选择来自于弹出式菜单的相应的项目。

6.8 直接查询导出

对于有巨大的结果集的查询（上万条记录或更多），首先查询数据到结果表格，随后把数据导出到文件里，这样做也许是不方便的。这也许会花很长时间和占用很多内存资源。这样还不如直接写结果集到导出文件里更有效。要做到这一点，你可以按 SQL 编辑器右边的*导出查询结果…*按钮，以取代工具栏里的*执行*按钮。这将显示出一个弹出式菜单，在这里你可以选择导出数据到 CSV 格式（逗号分隔值）、 TSV 格式（制表符分隔值）、 HTML 格式或 XML 格式。选择了格式以后，你就可以指定导出文件了，然后查询将被运行。在这种情形下，结果集不会被显示出来，只能被写到导出文件里。

6.9 保存 SQL 脚本

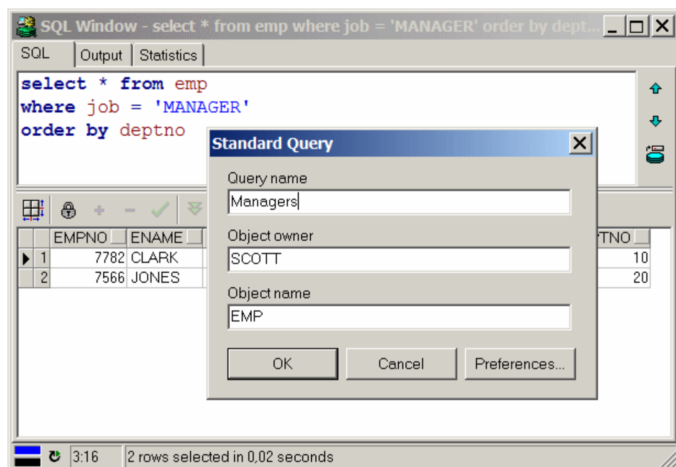
如果你已经创建了 SQL 语句，希望以后再次运行它，你就可以把它保存为 SQL 脚本。要做到这一点，请按工具栏里的*保存*按钮并且输入一个适当的文件名在保存对话框里。 SQL 脚本的默认扩展名为 *.sql* ，但是你可以使用第 14.26 章里描述的首选项来改变默认扩展名。注意，仅仅当前的 SQL 语句能被保存到文件里，并不是全部的历史语句。要把历史包括到保存文件里，请右击 SQL 窗口，选择*带有历史保存*项目。在那里，所有语句都将用斜杠分开。

你可以通过按工具栏里的*打开*按钮并选择 SQL 脚本来打开以前保存的 *SQL 脚本*，这将创建一个新的 SQL 窗口。作为一种选择，你可以右击以前创建的 SQL 窗口并选择*载入*项目，这将在现有的 SQL 窗口里打开一个 SQL 脚本。

6.10 创建标准查询

对于大多数表和视图，你会有各种各样的需要频繁运行的标准查询，例如：“显示所有是经理的雇员”或者“按部门号码显示所有雇员的工资总额”。你可以保存这样的查询为“标准查询”，以便以后能容易地从显示相应的表和视图的弹出式菜单里调用它。当你右击在对象浏览器里、或者在 PL/SQL 或 SQL 源里的表或视图，这个弹出式菜单就显示出来了。

在 SQL 窗口里创建了标准查询以后，你可以按结果表格工具栏里的*保存为标准查询*按钮，下列对话框就会显示出来：



你可以输入被显示在弹出式菜单里的查询的名字，你还可以改变所有者和将显示标准查询的表或视图的名字。按*首选项*按钮可以改变标准查询被存储的路径。

下列信息将与标准查询一起保存：

- SQL 文本
- SQL 窗口的大小
- 表格模式：多记录或单记录查看

注意，你可以使用置换变量（参看第 6.5 章）以使标准查询更为灵活。

7. 命令窗口

命令窗口允许你运行 SQL 脚本，这与 Oracle 的 SQL*Plus 很相似。要创建命令窗口，请按工具栏里的*新建*按钮或在*文件*菜单里选择*新建*项目。不用离开 PL/SQL Developer 的 IDE ， 一个命令窗口就被创建了，你可以键入 SQL 语句或者你习惯的 SQL*Plus 命令：

```
Command Window - New
Dialog | Editor
SQL> select * from emp
  2  where deptno = 20
  3  order by ename;

EMPNO ENAME      JOB         MGR HIREDATE          SAL       COMM DEPTNO
----- ---------- --------- ----- ----------- ---------- ---------- ------
 7876 ADAMS      CLERK      7788 12-01-1983    1100,00                20
 7902 FORD       ANALYST    7566 03-12-1981    3000,00                20
 7566 JONES      MANAGER    7839 02-04-1981    2975,00                20
 7788 SCOTT      ANALYST    7566 09-12-1982    3000,00                20
 7369 SMITH      CLERK      7902 17-12-1980     800,00                20

SQL> exec dbms_output.put_line(employee.deptname(7876))
RESEARCH

PL/SQL procedure successfully completed

SQL> desc dep
Object dep does not exist.

SQL> desc dept
Name   Type         Nullable Default Comments
------ ------------ -------- ------- ---------------------------------------------------
DEPTNO NUMBER(2)                     The 2-digit identification number of the department
DNAME  VARCHAR2(14) Y                The name of the department
LOC    VARCHAR2(13) Y                The location of the department

SQL> |
ECH TRM FDB VER APR HDG TMG  PL/SQL procedure successfully completed in 0 seconds
```

7.1 输入 SQL 语句和命令

就像在 SQL*Plus 里一样， 你可以键入多行 SQL 语句，用分号或者斜杠来结束输入。你可以使用左右箭头来编辑命令行，用上下箭头重新调用先前输入的行。

通过输入*编辑*命令，你可以使用简单的文本编辑器来编辑整个输入缓冲区。编辑了缓冲区以后，你可以通过在命令行上输入一个斜杠来运行它。编辑器有一个包括以前所有已运行命令的历史缓冲区，以至于你可以快速地运行已修改的命令。

命令窗口的状态行给出了显示命令、显示结果、反馈、校验、自动打印、标题和耗时选项。另外，你还可以通过双击这些选项来开和关它们。

7.2 开发命令文件

要开发带有多个 SQL 语句和命令的命令文件，你经常需要编辑这个文件，运行它，再编辑它、再运行它，等等。要把这些事情变成一个轻松的过程，命令窗口有一个带有 SQL 、 PL/SQL 和 SQL*Plus 语法高亮的内置的编辑器：

```
Command Window - demobld.sql
Dialog  Editor
--   This script creates the SQL*Plus demonstration tables in the
--   current schema.  It should be STARTed by each user wishing to
--   access the tables.  To remove the tables use the demodrop.sql
--   script.

SET TERMOUT ON
PROMPT Building demonstration tables.  Please wait.
SET TERMOUT OFF

--   Drop old tables
DROP TABLE EMP;
DROP TABLE DEPT;
DROP TABLE BONUS;
DROP TABLE SALGRADE;

--   Create new tables
CREATE TABLE EMP
       (EMPNO NUMBER(4) NOT NULL,
        ENAME VARCHAR2(10),
        JOB VARCHAR2(9),
        MGR NUMBER(4),
ECH TRM FDB VER APR HDG TMG
```

打开、执行和保存命令文件

要创建一个命令文件，请转到*编辑器*页并且输入命令。要在编辑器里执行命令，简单地按工具栏里的*执行*按钮或按 *F8* 就可以，命令窗口将转回到*对话框*页并且将执行所有命令。

要编辑一个现有的命令文件，请按工具栏里的*打开*按钮，选择*命令文件*项目。这将把命令文件载入到一个新的命令窗口的编辑器里。作为一种选择，你可以在编辑器里右击鼠标，选择来自于弹出式菜单的*载入*项目。要保存已修改的命令文件，请按工具栏里的*保存*按钮。

一步一步地执行命令文件

如果你在执行前或在执行期间按了状态行上的*暂停执行*按钮，你就可以以可控方式一步一步地运行脚本。下一个命令将被高亮显示在编辑器里，当你按状态行上的*执行下一个命令*按钮，它就将被执行了。如果你又按了*暂停执行*按钮，执行将在下一步以后正常地继续进行。你还可以通过使用 SET EXECPAUSE 命令在脚本里面强制设置一个暂停。

7.3 支持命令

除了所有的 SQL 语句之外，在这个版本里，命令窗口还支持下列标准的 SQL*PLus 命令：

命令	含义
/	运行 SQL 缓冲区
? [关键词]	对关键词提供 SQL 帮助
@[@] [文件名] [参数列表]	通过指定的参数，运行指定的命令文件
ACC[EPT] 变量 [DEF[AULT] 值] [PROMPT 文本 | NOPR[OMPT]]	允许用户输入置换变量的值
CL[EAR] [SCR[EEN]]	清除屏幕
CL[EAR] SQL	清除 SQL 缓冲区
COL[UMN] [列] [格式]	定义一个列的格式，显示一个列的格式，或者显示所有列格式
CON[NECT] [用户名/口令@数据库]	带有指定用户连接到数据库
DEF[INE] [变量] [= 文本]	定义置换变量，显示一个变量，或者显示所有置换变量。
DESC[RIBE] 对象	给指定对象一个描述
DISC[CONNECT]	从数据库断开
EDIT	显示文本编辑器以编辑 SQL 缓冲区
EXEC[UTE] 过程	运行指定的过程
EXIT [APPLICATION]	退出正在运行的脚本或关闭命令窗口，添加 APPLICATION 参数也将关闭 PL/SQL Developer 。
GET [文件名]	载入命令文件到编辑器
HOST [命令]	执行主机命令
HELP [关键词]	对关键词提供 SQL 帮助
PAUSE [信息]	显示信息和暂停，直到用户按确定或取消按钮为止
PRI[NT] [变量]	显示绑定变量或所有绑定变量的值
PROMPT [文本]	显示指定文本
QUIT [APPLICATION]	退出正在运行的脚本或关闭命令窗口，添加 APPLICATION 参数也将关闭 PL/SQL Developer 。
R[UN]	运行 SQL 缓冲区
REM[ARK] [文本]	一个注释行
SET AUTOP[RINT] [ON | OFF]	确定绑定变量在运行了 SQL 语句或 PL/SQL 块以后是否自动地被显示出来。
SET CON[CAT] [字符 | ON | OFF]	确定终止置换变量参照的字符（默认=.）
SET DEF[INE] [字符 | ON | OFF]	确定开始置换变量参照的字符（默认=&）

命令	含义
SET ECHO [ON | OFF]	确定在脚本里已执行的命令是否被显示出来
SET ESC[APE] [字符 | ON | OFF]	确定中断开始置换变量参照字符的字符（默认=\）
SET FEED[BACK] [ON | OFF]	确定 SQL 语句受影响行数是否被显示出来
SET HEA[DING] [ON | OFF]	确定在结果集列上面的标题是否被显示出来
SET LONG [宽度]	确定一个 LONG 列的最大显示宽度
SET NUM[WIDTH] [宽度]	确定不带精度的数字列的最大显示宽度
SET PAGES[IZE] [大小]	确定在标题被重复以前的结果集的显示行数
SET PROMPT [提示]	替换标准的 SQL> 提示，改为你也能使用变量 [user] 、 [db] 或 [connection] 的文字文本。此外，你还可以包括一个绑定变量（ set prompt :bind_var_name ）。
SET SCAN [ON | OFF]	确定置换变量是否应该被扫描
SET SERVEROUT[PUT] [ON | OFF] [SIZE n]	确定调用 dbms_output.put_line 的输出是否被显示出来，并且输出缓冲区的大小是多少。
SET TERM[OUT] [ON | OFF]	确定已运行的 SQL 语句的输出是否被显示出来
SET TIMI[NG] [ON | OFF]	确定关于已运行的 SQL 语句的计时信息是否被显示出来
SET VER[IFY] [ON | OFF]	确定当被用于 SQL 语句或 PL/SQL 块的时候置换变量是否被显示出来。
SHO[W] ERR[ORS] [类型名]	显示以前编译或者指定对象的错误
SHO[W] REL[EASE]	显示当前连接的 Oracle 版本信息
SHO[W] SQLCODE	显示已运行的 SQL 语句的结果代码
SHO[W] USER	显示当前连接的用户名
SPO[OL] [文件名 | OFF]	开始或停止假脱机
STA[RT] [文件名] [参数列表]	通过指定的参数，运行指定的命令文件
STORE SET [文件名]	在[文件名]里存储所有选项的值，你可以以后运行这个文件以恢复这些选项。
UNDEF[INE] 变量	不定义已设定的置换变量
VAR[IABLE] [变量] [数据类型]	定义一个绑定变量，显示一个绑定变量或显示所有的绑定变量。
WHENEVER [OSERROR | SQLERROR] [动作]	指定一个动作，用于无论 OS 或 SQL 何时出现错误。这个动作既可以是 EXIT 又可

命令	含义
	以是 CONTINUE ，随便地被 COMMIT 或 ROLLBACK 跟随着。

上面所有这些命令函数都与 SQL*Plus 相同。下面的命令是 PL/SQL Developer 特有的：

命令	含义
BEAUT[IFY] 文件 | 对象	使用当前规则或者 SET BEAUTIFIERRULES 命令指定的规则来美化指定文件或数据库对象。
BROWSE 对象	在对象浏览器里选择对象
EDIT 对象	打开带有对象定义的可编辑窗口
EDITD[ATA] 表 | 视图	为带有可编辑结果集的表或视图打开一个 SQL 窗口
EXPORT[DATA] 表	为指定表打开导出工具
INFO	显示关于连接的信息
PROP[ERTIES] 对象	为指定对象显示一个属性窗口
QUERY[DATA] 表 | 视图	为带有只读结果集的表或视图打开一个 SQL 窗口
REC[OMPILE] 对象	重新编译对象
SET BEAUT[IFIERRULES] [文件]	临时使用来自于指定文件的美化器规则，这可以带着 BEAUTIFY 命令使用。
SET COL[WIDTH] [宽度]	确定结果集里的最大列宽。如果 Width = 0 ，则不限制宽度。默认为 80 。
SET EXEC[PAUSE] [ON | OFF]	在下一个命令暂停执行（ ON ），或者在下一个命令正常继续执行（ OFF ）。
SQLPLUS	带有当前文件调用 SQL*Plus 。
TEST 程序单元	对于指定的程序单元，打开一个带有标准测试脚本的测试窗口。
VIEW 对象	打开一个带有对象定义的只读窗口

8. 创建与修改非 PL/SQL 对象

在 PL/SQL 开发期间，你会发现你经常需要创建表、修改约束或索引、复位一个序列等等。PL/SQL Developer 有几个功能允许你创建与修改表（以及相关元素）、序列、同义词、库、目录、用户和角色。这一章并不解释这些对象的功能，只解释你怎样才能创建、修改和查看这些属性。要了解关于每个对象的更多信息，请看 Oracle 的诸如*“ SQL 参考指南 ”*这样的文档。

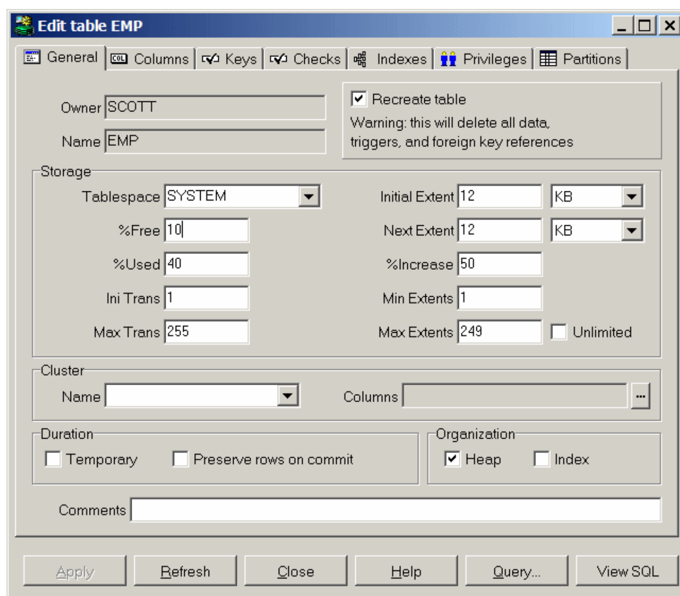
要创建一个对象，你既可以按工具栏里的*新建*按钮选择相应的对象类型，也可以选择*文件*菜单里的*新建*按钮。你还可以右击对象浏览器里对象类型的根文件夹，选择来自于弹出式菜单的*新建*项目。

要修改一个对象，就要在浏览器里选择它，右击它，选择来自于弹出式菜单的*编辑*按钮。你不能这样就改变一个对象的名字，必须明确地使用来自于对象浏览器的*重新命名*功能来改名（如果对象浏览器支持的话）。如果你仅仅要查看对象，就选择来自于弹出式菜单的*查看*项目。

另外，你还可以通过选择对象浏览器弹出式菜单里的*复制*项目来复制一个对象。这个对象的编辑器窗口将显示出来，它带有被填充在里面的、除了所有者（如果可应用的话）和名称之外的所有属性。这样，你能快速地创建一个相似的对象，以用于探索一个可供选择的办法、运行一些危险的或破坏性的测试、或者做其它任何事情。

8.1 表定义编辑器

表定义编辑器有 7 个标签页以用于表的不同方面：



在编辑器的底部，你能看到下列 6 个按钮：

- **应用　–　将应用你在编辑器里做的所有的修改到数据库里。**
- **刷新　–　再一次获取来自于数据库的定义，放弃你可能已经做了的任何改变。**
- **关闭　–　关闭编辑器窗口。**
- **帮助　–　显示在线帮助。**
- **查询　–　调用一个带有查询的 SQL 窗口，这个查询允许你查看和编辑表数据。**
- **查看 SQL　–　显示带有 SQL 语句的文本编辑器，这个 SQL 语句是表定义编辑器里产生了改变的结果。如果你还没有做任何改变，它将显示出完整的 SQL 创建语句。**

接下来的章节将对表定义编辑器的各个页进行描述。

一般页

***一般*页在先前的章节里有显示，它包括了表的所有者和名称、存储信息、簇信息和注释。对于一个新表，除了名称之外，你可以同意所有的属性为空。所有其它属性将设为默认值：**

- **所有者　–　当前登录的用户。**
- **表空间　–　当前用户默认的表空间**
- **%空闲　–　10**

- **%已用 – 40**
- **初始事务数 – 1**
- **最大事务数 – 255**

段属性的默认值（初始大小、下一个大小、%增加、最小数量和最大数量）依赖于相应的表空间默认值。

簇信息、存储信息和持续时间属性互相独占，这是因为簇会在暗中定义表存储特性，临时表不能成为簇并且也有固有的存储特性。

如果你输入或选择一个簇名，存储属性和持续时间属性将变为只读，簇列就能被输入。清除簇名字具有相反的作用。

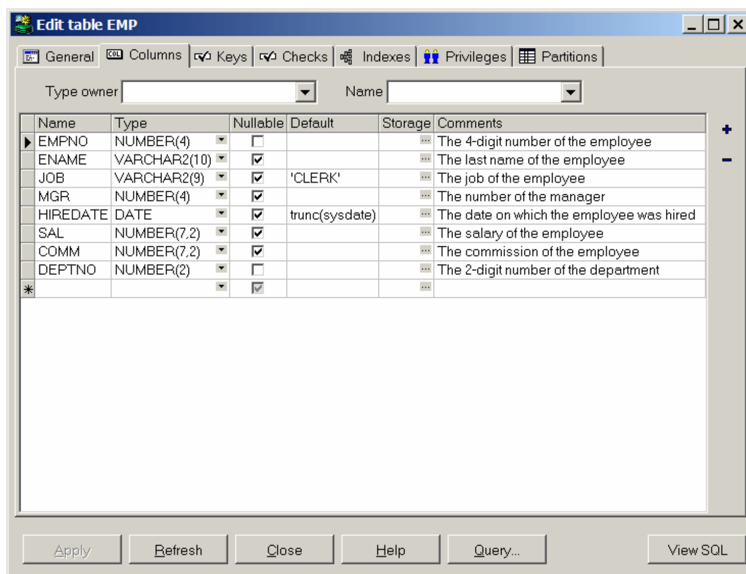
如果你定义表为临时的，簇和存储信息将变为只读，你就可以定义行在提交后是否需要被保护了。

如果你要定义一个索引结构表，你就必须定义一个主键。对于一个堆积构成的表（默认），这个就不需要了。

如果你正在修改一个现有的表，并不是所有的属性都能被改变。对于一个数据库里已经存在的表，表空间、初始大小、最小大小、簇信息、持续时间属性和构成属性都不能被改变。如果你要改变这些属性中的任何一个，你必须把窗口顶部的*重建表*选项设为已检验。结果，表首先被删除掉，再用新的定义重建。所有数据、触发器、外键参照<u>都会丢失</u>，所以在使用这个选项时，你要特别小心！

列页

在*列*页你可以查看、添加、删除、移动和修改表列。如果你创建了一个对象表，你还可以在这个页里选择对象类型：



要添加一个列，你可以按*插入列*按钮，或者你可以在末尾行（这里将创建一个新的空的末尾行）开始键入新列的信息。对于大多数通用数据类型，*类型*列有一个建议列表，但在这里你还可以使用其它数据类型。列的*默认*需要你按适当格式输入一个值，例如字符串值必须用引号引上。

要删除一个列，请按删除列按钮。这既会删除一个当前在焦点上的列，又能删除所有的已选列。你可以通过点击行标题选择一个或更多的列，拖动鼠标将选择一个范围的列。按 `Ctrl` 键同时点击行标题将允许你选择多个单独的列。

要移动一个或更多的列，就按前一段的描述选择它们。现在你就能点击已选标题中的一个，移动它到你希望的地方了。

对于现有的表，你不能改变现有列的名字，除非你用的是 Oracle9.2 或更新版本。你还不能删除一个这样的列，除非你用的是 Oracle8i 或者更新版本。要克服这些限制，你必须把*一般*页上的*重建表*选项设为已检验，这一选项的作用在先前的章节中有描述。

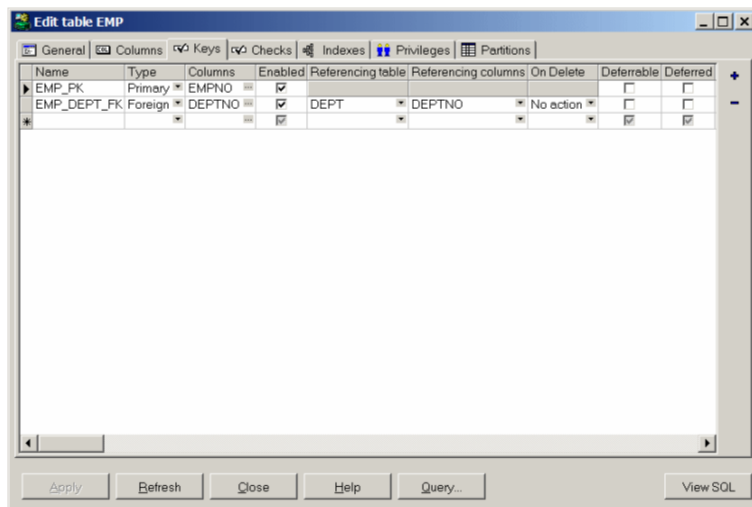
注意，改变列名将自动地被传递到约束列、索引列和簇列。

如果你在这个标签页的顶部选择了类型所有者和类型名，你就可以创建一个基于这个对象类型的对象表。现在列名和类型就被定义了，并且被限制到对象类型的顶级属

性。你还可以定义列是否允许为 null 、默认值、它的存储以及注释。如果你以后要改变对象类型，就必须选择*一般*页上的*重建表*选项。

键页

在*键*页上，你可以查看、添加、删除和修改表的主键、唯一键和外键约束：

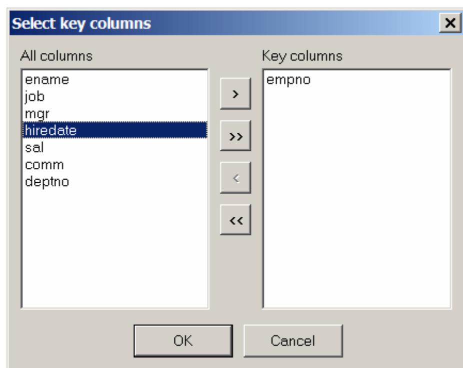


这个表格能按先前章节里列表格同样的方法操作。你可以添加、删除和修改键的约束。移动约束不会有任何用，就像它们没有被真正地排列好一样。

注意，被用于强迫为主键或唯一键约束的基础索引的存储信息在这个页里不被定义。如果约束被激活，在*索引*页上会自动有一个相应的具有相同名字的索引出现。对于这个索引，你可以定义存储定义。禁止或删除主键或者外键约束将在暗中删除基础索引。

*类型*列有一个允许你选择主、唯一或外键类型的列表。你可以通过键入首字符（ P 、U 或 F ）快速地选择恰当的类型。注意，表只能有一个主键约束，这个列（主或唯一）的默认值就依赖这个了。改变类型将影响*参照表*、*参照列*和*层叠*列，这是因为它们仅仅对外键约束有效。改变类型到外键也将暗中删除索引。

要定义*列*到约束应用的地方，你在适当的单元格（被逗号分隔）里简单地键入列名就可以，或者按单元格按钮也可以。这将调出列选择屏幕，在这里你能容易地添加、移除或移动（重排）键列：

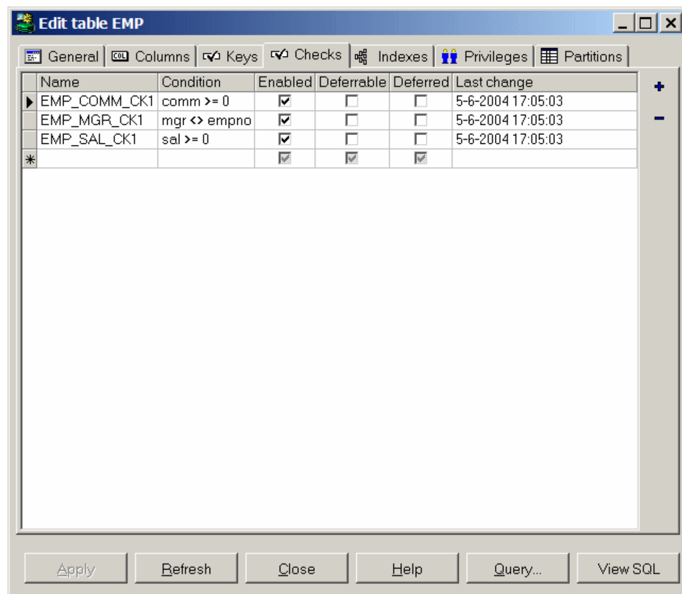


对于外键约束，你可以键入或者选择*参照表*。这样做将自动地选择基于这个表主和唯一键约束的默认的*参照列*。这个列的选择列表显示出所有列的设置，它们为参照表整理出主或唯一键约束，因此在这里它们仅仅为候选者。在*级联删除*列表栏里，对于父记录被删除时子表里的记录，你可以定义应该执行什么行动。*Set null* 行为在 Oracle8i 和更新的版本里被支持。

如果你用的是 Oracle 8i ，你还可以定义约束是否可延迟，在最初是否已延迟。这两个选项有明显的关系：你不能拥有一个初始已延迟的而又是一个不可延迟的约束。

检查页

在*检查*页里你可以查看、添加、删除和修改表的检查约束：



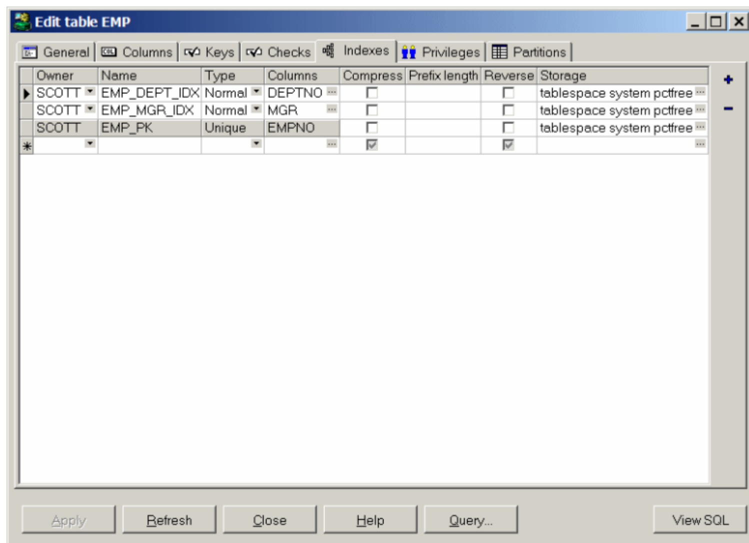
这个表格能用与列表格一样的方法操作，你可以添加、删除和修改检查约束。

***条件*列应该包括了对表里的所有行都必须是真（或空）的布尔表达式。如果一个或多个行违反条件，你就不能应用检查约束，除非你同样不设*允许*选项为已检验。在改正了违反条件行以后，检查约束才能被允许。**

如果你用的是 Oracle8 或更新版本，你还可以定义约束是否是可延迟的、最初是否被延迟。这两个选项有明显的关系：你不能拥有一个初始已延迟的而又是一个不可延迟的约束。

索引页

在*索引*页里你可以查看、添加、删除和修改表的索引：



这个表格能用与列表格一样的方法操作，你可以添加、删除和修改索引。

索引的*所有者*能被设为空，默认的所有者是当前登录的用户。如果你要为索引使用不同的用户，就从这个列表里选择它。

索引的类型可以是 Normal 、Unique 或 Bitmap 。最后一个索引类型只能用于 Oracle7.3 服务器和更新版本。注意，如果数据库里每一行的索引列都是真正唯一的，你就只能创建唯一索引。只要不是这种情况，你就不能应用这个索引到数据库里。

要定义你要建索引的表的*列*，你在适当的单元格（被逗号分隔）里简单地键入列名或者按单元格按钮就可以。这将调出列选择屏幕，在这里你能容易地添加、移除或安排索引列。

对于基于索引的函数，你可以指定函数，用逗号分隔。对于递减索引，你可以用 *DESC* 跟随列名。

你还可以允许索引的键压缩，包括前缀列的数字压缩，你还可以定义索引为相反的索引。

如果索引是主或唯一键的索引，则所有者、名称、类型和列就不能被修改。这些特性全部得自键约束。改变键约束的名称将自动地改变相应的索引名，改变约束列将改变用于索引的相应的列。索引的所有者总是与表的所有者相同，索引的类型总是唯一的。

*存储*参数能通过按单元格按钮来定义或查看，这将调用存储编辑器：

Storage parameters

Tablespace	USER_DATA	Initial Extent	10	KB
%Free	10	Next Extent	10	KB
%Used		%Increase	50	
Ini Trans	2	Min Extents	1	
Max Trans	255	Max Extents	121	Unlimited

OK Cancel

对于新索引，你不必输入任何信息，因为每一个属性都有一个默认值：

- 表空间 – 针对当前用户默认的表空间。
- %空闲 – 10
- 初始事务数 – 2
- 最大事务数 – 255

段属性的默认值（初始大小、下一个大小、% 增加、最小数量和最大数量）依赖于相应的表空间默认值。% 已用特性不能应用到索引里。

注意，如果你修改了现有索引的 % 空闲、初始大小和最小数量特性，这将暗示索引（和约束）将被删除掉和被重新创建，这种改变也会被应用。对于大的索引，这可能是非常耗费时间的。

权限页

在权限页里，你可以在表里授予/撤回给/来自用户和角色的权限：

Grantee	Select	Insert	Update	Delete	References	Alter	Index
DEMO_SELECT	Yes						
DEMO_UPDATE	Yes	Yes	Yes	Yes			
KING	Grantable	Yes	Yes	Yes	Yes	Yes	Yes
PUBLIC	Yes						

这个表格能用与列表格一样的方法操作，你可以添加、删除和修改被授予者。删除一个被授予者与撤回这个人所有的权限等效。

***被授予者*是你要授予权限的用户或角色。注意，给角色授予的权限常常不能满足于存储程序单元开发的目的。如果用户在表里通过角色被授予了 select 权限，参照这个在已存储的程序单元里的表，编译将失败，这是因为表是未知的。如果对象是被用于他或她拥有的程序单元里的，用户必须亲自被授予 select 权限。**

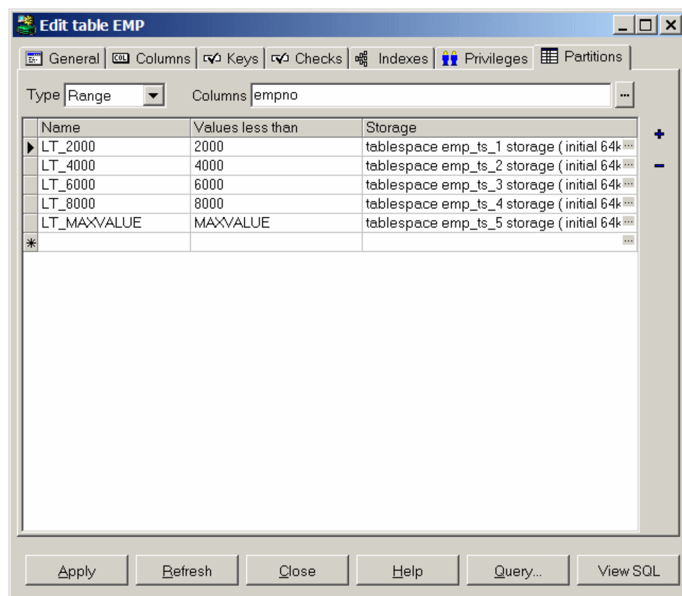
单独的权限列能有 3 个值：

- **（空白） – 被授予者没有权限**
- **Yes – 被授予者有权限**
- **Grantable –被授予者有权限，并且能授予它到其它用户和角色**

***选择*、*插入*、*更新*、*删除*权限允许被授予者在表记录里运行这些动作，*参照*权限允许被授予者创建外键到这个表，*改变*权限允许被授予者用改变表命令改变表定义，*索引*权限允许被授予者在表里创建索引。**

分区页

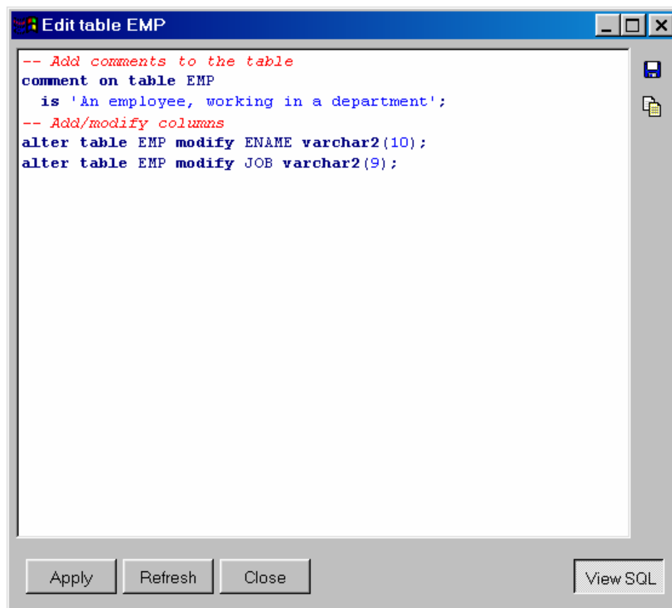
分区页仅仅在你（重新）创建表的时候或者在你编辑现有的已分区的表的时候是可见的：



在页的顶部，你可以选择 *Range* 、 *Hash* 或 *List* 来分区，使用分区列也可以。在表格里，你可以定义名称、列的值（如果可应用的话），可以存储每一个分区。如果你定义了不只一个分区列，你必须用逗号来分隔这些值。

查看与修改 SQL

在创建了一个表以后，或者修改了一个现有的表以后，你可以通过按*查看 SQL* 按钮来查看作为结果的 SQL 。在添加了注释到表中并且修改了 *ename* 和 *job* 列到 varchar2 数据类型后，作为结果的 SQL 就像这样：



你可以使用窗口右上角的相应按钮保存 SQL 到一个文件或者复制它到剪贴板，工具栏里的*打印*按钮将打印 SQL 。你还可以产生一个改变到 SQL 并且应用已改变的 SQL 。在应用了已改变的 SQL 以后，窗口将转到表格模式并且刷新表定义。已产生到 SQL 的改变将立即在表格里被反映出来。

如果你再一次按*查看 SQL* 按钮，窗口将转到表格模式，抛弃了任何你可能已经生成到 SQL 的未应用的改变。

如果你已经选择了浏览器里的*查看*选项来查看表定义，*查看 SQL* 按钮就将显示表创建的 SQL ，不带有前面所述的带有删掉命令的它。这样，你可以在文件里保存表定义，或者复制表定义到另一个数据库或用户。

8.2 序列定义编辑器

序列编辑器允许你容易地创建或修改一个序列：

大多数属性都有一个默认值，当创建一个序列时，你只需要应用一个名字：

- **所有者 – 当前登录的用户**
- **最小值 – 1**
- **最大值 – 不限制**
- **初始值 – 1**
- **增量 – 1**
- **高速缓冲区大小 – 20**
- **循环 – 关**
- **排序 – 关**

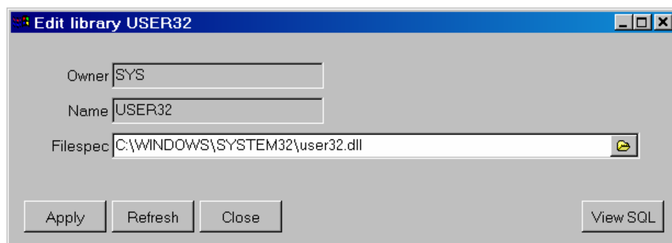
8.3 同义词定义编辑器

同义词编辑器允许你容易地创建或修改同义词：

***所有者*的默认值将是当前登录的用户。设*公共*选项为已检验将创建公共同义词，并且将产生只读的*所有者*输入框。*对象所有者*和*对象名称*输入框有一个建议列表，以便容易地为同义词选择一个事务对象。如果*对象所有者*输入框是空的，*对象名称*列表将显示出所有对象。**

8.4 库定义编辑器

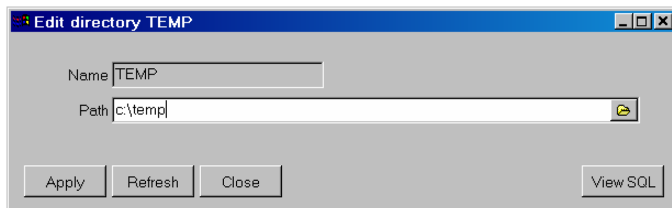
库定义编辑器允许你容易地创建或修改库：



_所有者_的默认值将是当前登录的用户，你可以使用建议列表来创建带有不同所有者的库。_文件说明_是你要使用的 3GL 库完整的文件说明。如果你正在数据库服务器上搞开发，那么你可以使用_选择 DLL 文件_按钮来选择动态连接库。另外，你必须键入 DLL（Windows）的名字或共享对象（UNIX）。文件说明的有效性并不被检验。

8.5 目录定义编辑器

目录编辑器允许你容易地创建或修改一个目录，这个目录随后能被用于 BFILE ：



如果你正在数据库服务器上搞开发，那么你可以使用_选择目录_按钮来选择目录的路径。目录路径的有效性并不被检验。

8.6 作业定义编辑器

作业定义编辑器允许你创建或修改一个作业：

创建一个新的作业与调用 dbms_job.submit 相同，修改一个现有的作业与调用 dbms_job.change 相同。要了解更多关于作业的信息，请看 *Oracle 供应包参考*手册里的 *DBMS_JOB* 章。

What **输入区域旁边的*打开 PL/SQL 编辑器*按钮可以调用更加方便的带有语法高亮的 PL/SQL 编辑器、代码助手，等等。**

8.7 队列定义编辑器

队列定义编辑器允许你创建或修改一个高级队列：

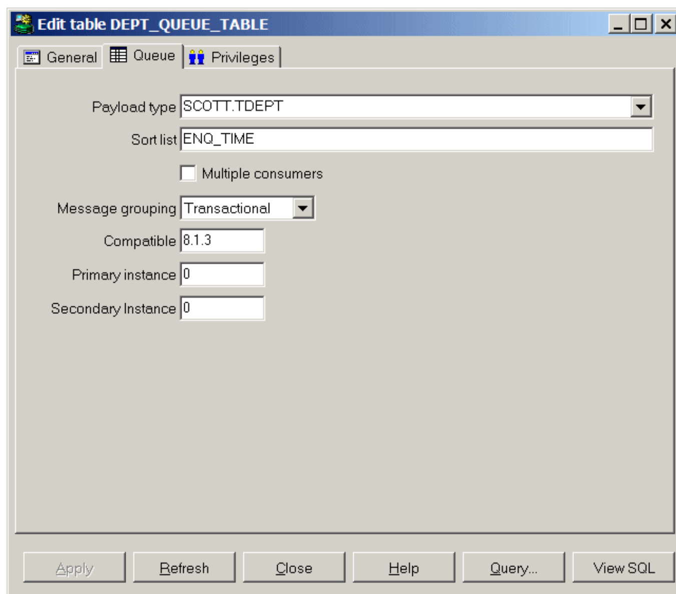
对于队列表的名称，你可以选择一个现有的队列表，或者你可以按*创建*按钮来创建一个新的队列表（ 请看第 8.8 章）。

要了解关于 Oracle 高级队列的更多信息，请看下列 Oracle 文档：

- ***应用程序开发指南 – 高级队列***
- ***补充的 PL/SQL 包和类型参考***

8.8 队列表定义编辑器

队列表定义编辑器允许你创建或修改一个高级队列表：



在*一般*标签页上，你可以定义表的名称、存储参数和注释。在*权限*标签页上，你可以在表上授权给用户和角色。请看第 8.1 章。

在*队列*页上，你可以提供队列表说明信息。对于*有效载荷*，你可以选择一个现有的对象类型或 *RAW* 来创建一个裸队列。

要了解关于 Oracle 高级队列的更多信息，请看下列 Oracle 文档：

- *应用程序开发指南 – 高级队列*
- *补充的 PL/SQL 包和类型参考*

8.9 用户定义编辑器

用户定义编辑器允许你创建或修改一个用户、它的权限和表空间限额：

注意，你需要指定系统权限（例如由 DBA 角色提供的）来查看和修改用户信息。

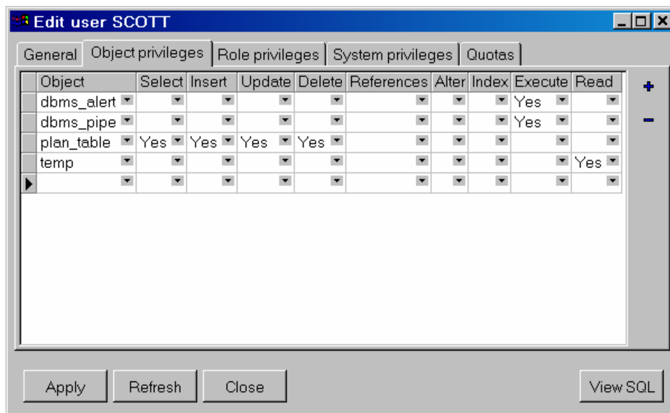
一般页

在一般页里，你基本上只需要输入名字和口令，所有其它属性都有一个默认值：

- 默认表空间 – SYSTEM
- 临时表空间 – SYSTEM
- 概要文件 – DEFAULT
- 口令期限 – 关
- 账户被锁住 – 关

如果你要修改一个现有的用户，当前的口令将不被显示。输入一个新的口令可以改变它，或保留为空白则不改变口令。如果用户要在外表上被鉴别，你就不能输入一个口令。

对象权限页

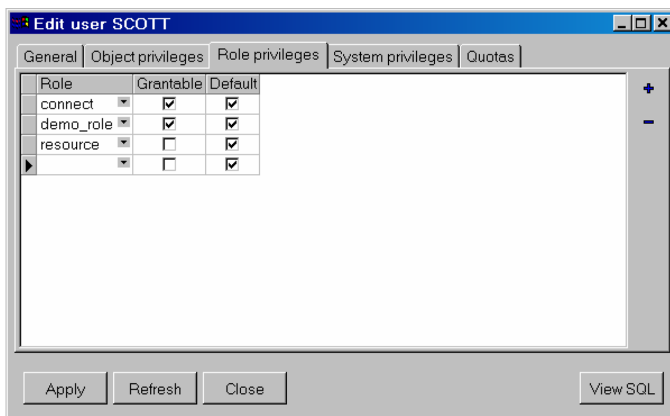


这一页允许你查看和修改被授予到用户的对象权限，这不包括间接地通过角色授予的权限。

在第一列里，你可以看到被授权到用户的对象。所有其它列都描绘出了一个指定的权限，每一个权限可以有下列值：

- （空白） – 用户没有权限
- Yes – 用户有权限
- Grantable –用户有权限，并且能授予权限到其它用户或角色。

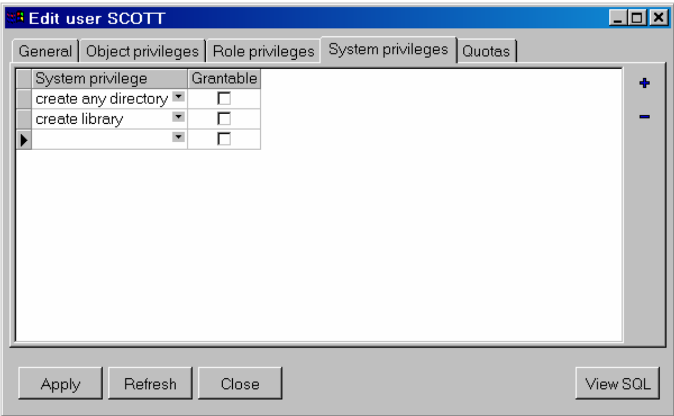
角色权限页



这一页显示了被授权到用户的角色。每一个角色权限都是*可授予*的，所以用户可以授予权限到其它用户和角色。如果角色权限是*默认*的，则用户登录时角色将被激活；如

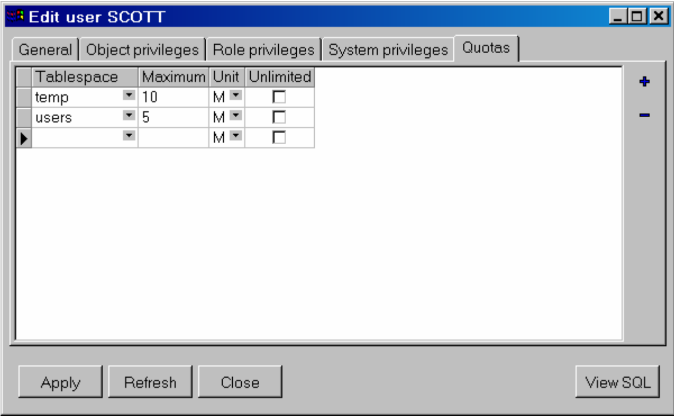
果它不是默认的，则对于在用户已经登录后通过执行 *set role* 命令的会话，角色权限必须明确地被激活。

系统权限页



这一页显示了被授予到用户的系统权限。每一个系统权限都可以是可授予的，所以用户能授予它到其它用户和角色。

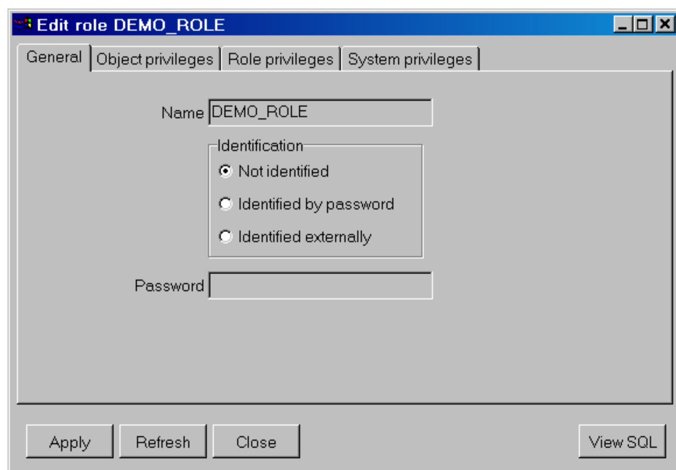
限额页



这一页显示了用户的表空间限额。你可以输入用户能在表空间里分配的最大数量的字节、千字节或兆字节，或者你可以指定限额为*不限制*。

8.10 角色定义编辑器

角色定义编辑器允许你创建或修改一个角色和它的权限：



在默认的状态下，角色不需要被鉴别。你同样也可以对角色定义口令鉴别或外部鉴别。口令鉴别需要你输入一个口令，如果角色没有默认地被授权到用户，这就是有用的。在用户登录以后，这样的角色需要被明确地设置，并且需要口令。

其它三个标签页（对象权限、角色权限和系统权限）以与用户定义功能相应页一样的方法工作。

8.11 概要文件定义编辑器

概要文件定义编辑器允许你编辑资源限制、口令限制和用户的概要配置：

每一个限制能被设置为*默认*，这样值将从标准概要配置的 DEFAULT 到 *Unlimited* 或一个具体值被继承。

在用户标签页上，你可以定义到应用这个概要文件的哪一个用户。注意，如果你从概要文件里移除一个用户，它将在暗中被改变到 DEFAULT 概要文件里。

8.12 数据库连接定义编辑器

数据库连接定义编辑器允许你创建和编辑数据库连接：

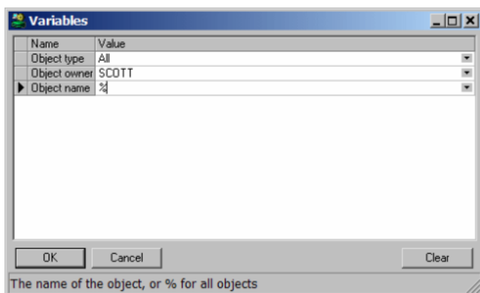
注意，当你编辑一个现有的数据库连接时，*口令*和*鉴别*属性将不会被获取，你必须在编辑数据库连接时每次都输入这个信息。

9. 报告

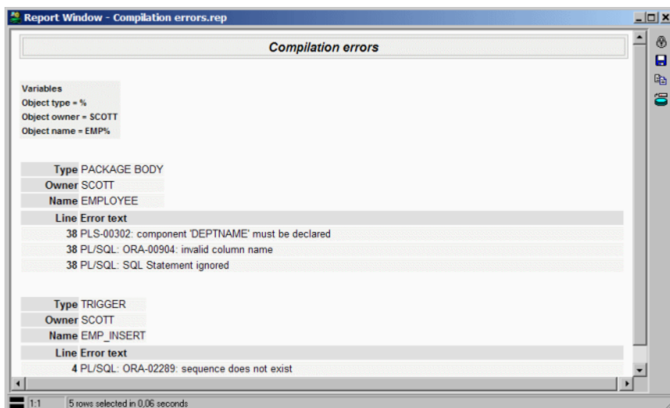
要运行面向你的应用数据或 Oracle 词典的报告，你可以使用 PL/SQL Developer 内置的报告功能。这里有很多标准的报告，并且你还可以创建你自己的定制报告。这些定制报告能被保存在报告文件里，它们能依次被包括在报告菜单里，这对你经常运行自己习惯的定制报告是很容易的。

9.1 标准报告

标准报告通过*报告*主菜单是可以访问的。例如，如果你选择了*编译错误*报告，你将首先被提示输入你要显示当前编译错误的对象类型、所有者和名称：



在输入了适当的值并按了*确定*按钮后，报告将被显示出来：



其结果是一个 HTML 文档，可以通过 PL/SQL Developer 内部的 HTML 查看器或者 Internet Explorer ActiveX 控件来显示，这两种显示方法的选择依赖于首选项（请看第 14.8 章）。

随后你可以通过按工具栏里的*打印*按钮来打印报告，或者你可以通过按报告窗口右边的*保存结果*按钮来保存 HTML 格式的报告。*复制为 HTML* 按钮将复制结果到剪贴板。

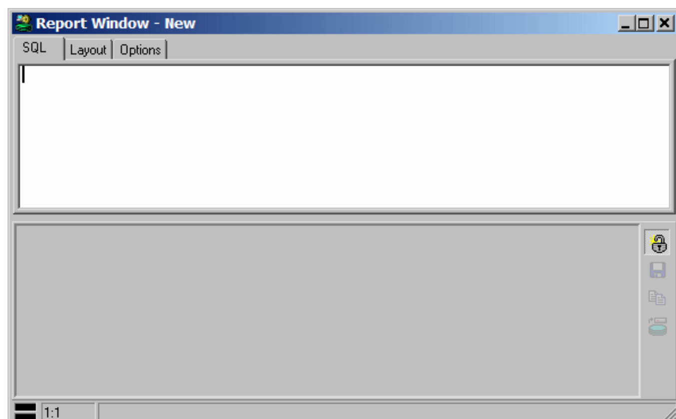
导出结果按钮允许你导出结果为 CSV 、 TSV 或 XML 格式，或者直接导出结果到 Excel 。

如果报告没有被锁定，***编辑报告***按钮提供了对报告定义的访问。

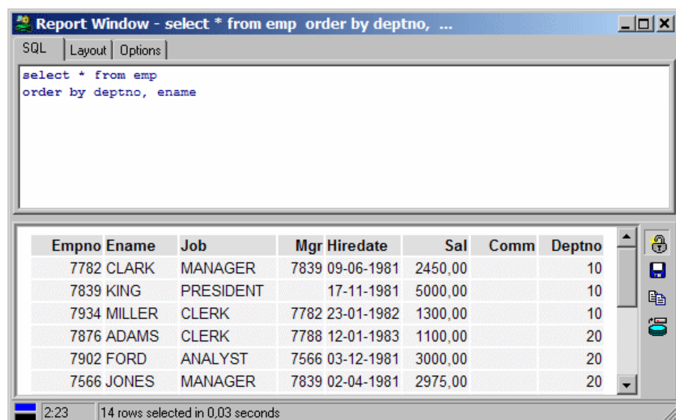
你可以右击报告来选择另外的选项。

9.2 定制报告

创建定制报告是很容易的。首先你通过工具栏里的新建按钮创建一个新的空白报告，然后选择来自于弹出式菜单的***报告窗口***项目。一个空的报告窗口将显示出来：



在 *SQL* 编辑器里，你可以为你的报告键入一个简单的 SQL select 语句。对于主题/明细报告，你可以使用内部联接的 select 语句（这将在以后描述）。让我们通过键入 select 语句和按工具栏里的***执行***按钮，简单地以清楚明白的 *emp* 表来开始：



这是一个很基本的报告，仅仅带有已应用的默认字体属性。下面章节将描述你怎样才能精制报告功能。

9.3 变量

在你可以运行前，你的报告经常需要某些额外的输入。让我们来假定你要限定来自于先前章节的报告到仅仅一个部门的雇员。如果是那样的话，你需要在 SQL 文本里提供一个置换变量：

```
select * from emp
where deptno = &department
order by ename
```

当这个报告被运行时，在 select 语句被运行前，用户会被提示给 *department* 输入一个值。这当然并不十分安全。要是用户什么都不输入呢？这个语句将会导致 *ORA-00936：缺少表达式* 异常。要是用户输入的值不是数字呢？或者要是输入的数字在 *dept* 表里不存在呢？要防止这些情况，你可以使用下面章节中描述的参数选项。这些选项被指定在两个括号之间，就像 HTML 或 XML 文档里的属性一样。

Name 选项

在变量输入表里，变量的名称被作为提示来使用，所以你应该确认它确实描述了变量的含义。如果名称仅仅是一个选项，那么它仅仅是一个不必在两个括号之间指定的选项。如果名称包括了空格或其它特殊字符，那么就用双引号引上它。下面的两个例子是等效的：

```
select * from emp
where deptno = &"Department number"
order by ename
```

和

```
select * from emp
where deptno = &<name="Department number">
order by ename
```

Hint 选项

除了名称选项之外，你还可以指定提示文本。当变量有焦点的时候，这个提示将被显示在变量输入表的状态行上：

```
select * from emp
where deptno = &<name="Department number"
                 hint="Only employees from this department will be listed">
order by ename
```

Type 选项

类型选项限定用户能被输入的信息，也控制什么样的值应该被插入到作为结果的 SQL 文本：

```
select * from emp
where deptno = &<name="Department number"
                 type="integer">
order by ename
```

在这种情形下，用户只能输入有效的整数值，输入的值将原封不动地被放到 SQL 文本里。然而对于字符串，你要把值用引号引上。考虑一下下列的 select 语句：

```
select * from emp
where ename = &<name=”Employee name”
                type=”string”>
```

如果你在变量输入表里输入 *SCOTT* （没有引号），作为结果的 SQL 将是：

```
select * from emp
where ename = ‘SCOTT’
```

有效的类型是： *none* 、*string* 、*integer* 、*float* 和 *date* 。对于日期值，有效将不出现，所以用户可以另外提供诸如 *sysdate* 、*trunc(sysdate, ‘MM’)* 等表达式。

Required 选项

对于某一变量，如果你要求有一个值，那么就设它的 *required* 选项为 *“yes”* 或 *“true”*：

```
select * from emp
where deptno = &<name=”Department number”
                 required=”yes”>
order by ename
```

现在，对于部门号码，如果一个值被指定，那么报告只能被运行。

Default 选项

要为变量提出一个默认值，就要指定 *default* 选项：

```
select * from emp
where deptno = &<name=”Department number”
                 default=”10”>
order by ename
```

默认值也可以是一个 select 语句，这个 select 语句应该仅仅能返回一行一列。例如：

```
select * from emp
where deptno = &<name=”Department number”
                 default=”select min(deptno) from dept”>
order by ename
```

在这种情况下，默认值将是最小的部门号码。

Ifempty 选项

作为一种选择，要生成一个需要的变量或提供一个默认的值，万一用户不指定它，你可以指定 *ifempty* 值：

```
select * from emp
where ename like &<name=”Employee name”
                   type=”string”
                   ifempty=”%”>
```

如果用户不指定一个值，百分号就会被使用，所有的雇员将会被获取。这个选项不能与 *required* 选项一起被指定。

Uppercase 选项

如果你有一个字符串变量要在暗中把它的值转换到大写，那么就设置它的 *uppercase* 选项为“*yes*” 或“*true*”：

```
select * from emp
where ename = &<name="Employee name"
                type="string"
                uppercase="yes">
```

值作为形态被显示，但在作为结果的 SQL 文本里被转换为大写。

Readonly 选项

变量可以通过指定 readonly="yes" 被定义为只读。用户可以看到变量和它的值，但不能改变它。

Hidden 选项

你可以通过指定 hidden="yes" 来隐藏变量。

List 选项

list 选项为用户提供了可能的值的列表，你可以指定项目以逗号分隔的列表：

```
select * from emp
where deptno = &<name="Department number"
                 list="10, 20, 30, 40">
order by ename
```

这允许用户选择四个项目中的一个，或键入一个不同的值。在这种情况下，使用 select 语句会有更多的意义：

```
select * from emp
where deptno = &<name="Department number"
                 list="select deptno from dept order by deptno">
order by ename
```

你还可以为每一个项目提供一种描述：

```
select * from emp
where deptno = &<name="Department number"
                 list="10, ACCOUNTING,
                       20, RESEARCH,
                       30, SALES,
                       40, OPERATIONS"
                description="yes">
order by ename
```

现在每一个项目都被描述跟随着，并且 *description* 选项被添加了。这个列表仅仅显示描述，实际值将被用于作为结果的 SQL 文本。

此外，你还可以用两个字段的 select 语句做同样的事情：

```
select * from emp
where deptno = &<name="Department number"
                 list="select deptno, dname from dept order by dname"
                 description="yes">
order by ename
```

要限定用户到列表里正确的项目，就要指定 *restricted* 选项：

```
select * from emp
where deptno = &<name=”Department number”
                 list=”select deptno, dname from dept order by dname”
                 description=”yes”
                 restricted=”yes”>
order by ename
```

现在，用户只能从列表里选择值了，不能手工输入任何其它值。

列表里的项目有时能依赖另外一个变量的值。假设你要创建显示指定表所有列的报告，表通过所有者和名称来鉴别，对于所有者你可以使用数据库里所有用户的列表，对于表名你可以查询已选所有者的 *all_tables* 视图：

```
select * from all_tab_columns
where owner = &<name="Owner"
                type="string"
                list="select username from all_users order by username">
and table_name = &<name="Table"
                   type="string"
                   list="select table_name from all_tables
                         where owner = :owner
                         order by table_name">
order by column_id
```

正象你可以看到的，第二个（表）列表的 select 语句通过 *:owner* 绑定变量引用了第一个变量。只要 owner 的值被改变，表列表就将被移民。

注意，绑定变量的名称不能长于 30 个字符，不能包含空格或其它特殊字符。变量名里的空格将被转换为下划线。如果变量名是 *Owner of the table* ，则绑定变量名将是 *:owner_of_the_table* 。

还要注意的是，如果变量的值是空的，则依赖的列表也将是空的。在这种情形下，查询不会被运行。

Checkbox 选项

如果用户选择两个可能性中的一个，你可以使用 *checkbox* 选项。这个选项需要有已检验和未检验值跟随：

```
select * from emp
order by hiredate &<name=”Descending sort order”
                    checkbox=”desc, asc”>
```

即使你总是必须指定两个值，其中之一也还是可以为空：

```
select * from emp
order by hiredate &<name=”Descending sort order”
                    checkbox=”desc,”>
```

因为都默认为上升排序，所以这两个例子是等效的。

Prefix 和 suffix 选项

如果变量值是空的，这可能意味着 SQL 文本的某一固定部分也必须被省略。考虑一下你要提供一个可选择的排序列的例子，如果列没有被指定， *order by* 子句应该

被完全忽略。在这种情况下，你可以指定这个文本为变量的 *prefix* 或 *suffix* 。如果变量值是空的，前缀和后缀还是被忽略为好：

```
select * from emp
&<name=”Descending sort column”
  list=”empno, ename, hiredate”
  prefix=”order by "
  suffix=” desc”>
```

如果这个变量的值没有被指定，那么结果还是为空好：

```
select * from emp
```

如果列被指定，结果应该是：

```
select * from emp
order by empno desc
```

要注意前缀后面的和后缀前面的空格，这些是重要的，因为前缀、值和后缀都象被指定那样紧密地连接在一起！

换码符

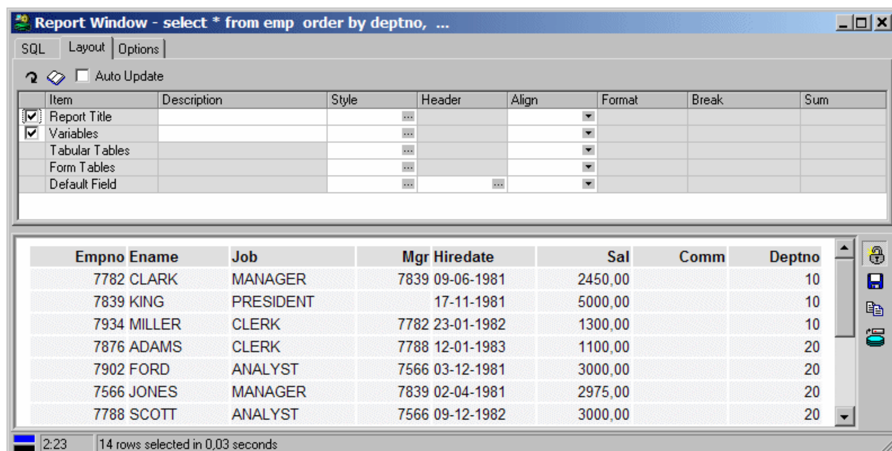
如果你希望在 SQL 文本里使用应该不被解释为置换变量的 & 符号，那么就用双 & 符号来代替。下面的例子将获取来自于‘R&D’部门的所有雇员：

```
select * from emp
where emp.deptno in (select dept.deptno from dept
                     where dname = 'R&&D')
order by empno desc
```

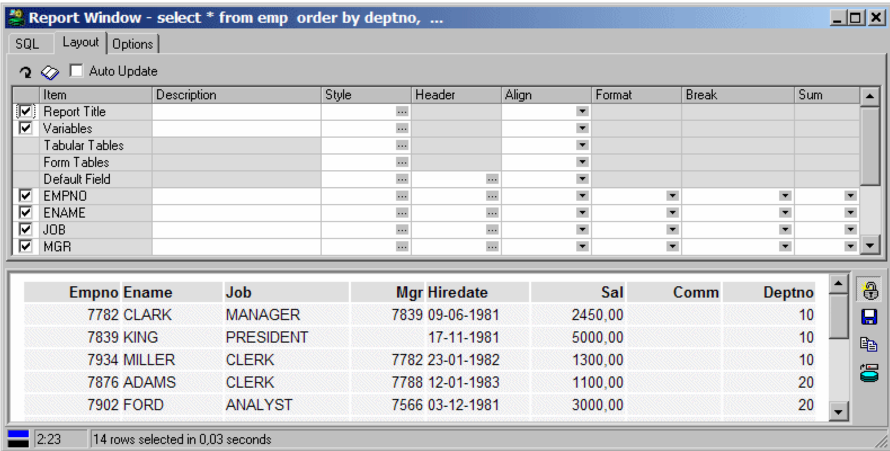
如果代之以使用文本‘R&D’，你就会因 D 变量而被提示。

9.4 精制版面

打开了一个新的报告窗口以后，键入一个 select 语句，再运行报告，版面将使用默认的风格属性。它将有一个简单的表格式版面，带有标准的字体属性、表风格、颜色等等。要精制标准的版面，请转到*版面*标签页。对于新的报告，这一页将如下所示：



正如你看到的，你可以为所有版面项目（报告标题、变量、表和字段）定义各种版面属性。如果你要为一个单独的字段定义属性，你首先必须按工具栏里的*刷新字段列表*按钮：

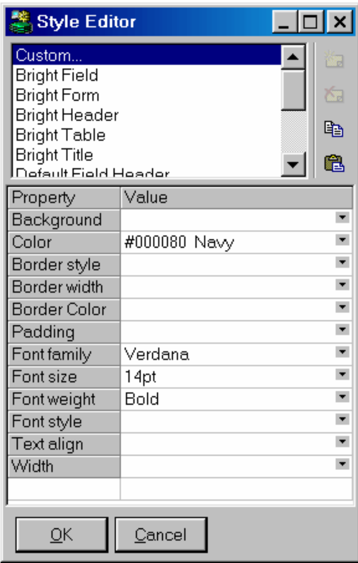


现在，select 语句结果集的所有字段都被包括到版面表格里了，你可以设置每一个单独字段的版面属性。如果你把字段的*风格*、*标题*或*对齐*保留为空，那么来自于*默认字段*的相应属性将被使用。

改变了版面属性以后，你可以再一次运行报告来查看结果的效果。作为一种选择，你可以激活工具栏里的*自动更新*选项，这样每一个改变都会马上反映到结果里。

下面的章节详细地描述了各种各样的版面属性。

显示



版面表格最左边一列的检验栏指出版面项目是否应该被显示，这允许你禁止报告标题或变量，允许你控制指定字段是否应该被显示。

描述

*报告标题*的描述将在报告的顶部和在每一个分页符之后被显示出来，*变量*的描述将被显示在变量的上面。

对于单独的字段，你可以越过标准的字段名来定义一个描述，这个描述将用于列标题。

风格

风格控制了版面项目的外观，按单元格按钮（...）就调出了风格编辑器：

在风格编辑器顶部的列表栏里，你可以看到所有的标准风格和*定制*风格。你可以从风格库（请看第 9.5 章）里选择一个标准的风格，或者你可以选择定制的风格以便你能为当前的版面项目改变各种风格属性。

注意，这个风格系统是基于层叠式样式表（CSS）的标准。你在风格编辑器里看到的属性是最常用的，当然你还可以在标准的属性下面添加其它的 CSS 属性和值。例如，你可以添加一个值为 *20pt* 的 *Height* 风格属性，它把单元格高度设置为 20 个点。关于层叠式样式表的更多信息，请看 http://www.w3.org/TR/REC-CSS2 。

*复制*与*粘贴*按钮允许你复制一个风格的属性，所以你可以快速地得到一个来自于现有风格的新的风格。

如果你没有为字段风格定义什么，来自于风格库的 *Default Field* 风格将被使用。你可以通过定义 *Default Field* 版面项目的风格来越过这个，你可以在字段级通过指定相应的风格来越过这个默认风格。注意，每一个字段标题继承了它的来自于 *Default Header* 风格的风格属性，所以你只需要越过你要改变的属性。例如，如果你定义了 *Default Field* 风格为 Verdana，14 pt，blue ，并且你要显示的 *EMPNO* 列为 Verdana，14 pt， red ，那么你只需要在 EMPNO 级定义颜色。字体和字号将自动地从 *Default Field* 级被继承。

各种版面项目的默认风格在第 9.5 章里有描述。

标题

正如你能为个别字段数据设置风格，你还能为它们的标题设置风格。如果你没做任何定义，来自于风格库的 *Default Field Header* 风格将被使用。你可以通过定义 *Default Header* 版面项目的风格来越过这个，你可以通过指定相应的标题风格在字段级越过这个默认风格。注意，每一个字段标题都继承了它的来自于 *Default Header* 风格的风格属性，所以你只需要越过你要改变的属性就可以。

对齐

你可以按下列方法对齐版面项目：

- 左 – 项目在单元格里左对齐
- 右 – 项目在单元格里右对齐
- 居中 – 项目在单元格里居中对齐
- 默认 – 数字右对齐，其它项目左对齐
- 无 – 不指定对齐

注意，你还可以为版面项目设置*文本对齐*风格属性，这将把先后次序带到你可以在版面表格里指定的对齐选项。还要注意的是，对齐的效果可能依赖于版面项目的 *Width* 风格属性。

格式

在默认的情况下，字段的格式被数据类型、数值范围和精度所控制。你常常要使用被计算的字段或合计的字段，这样数值范围和精度是未知的。在这种情况下，你可以指定一个格式。

格式属性仅仅对日期和数字字段有效果。对于数字字段，你可以使用 9 为禁止零的阿拉伯数字、0 为正常的阿拉伯数字、G 为千分隔符、D 为十进制分隔符、E 为科学符号。另外，你还可以在单引号或双引号里指定文字文本。一些例子：

格式	值 = 1234	值 = 5	值 = 0.1
999G990D00	1,234.00	5.00	0.10
9G990	1,234	5	0
0000	1234	0005	0000
0”%”	1234%	5%	0%
0D000e+00	1.234e+03	5.000e+00	1.000e-01
0D000e-0	1.234e3	5.000e0	1.000e-1

对于日期字段，你可以使用标准的 Windows 日期和时间格式。

中断

中断版面属性允许你构成你的报告结果。假定你要显示部门和它们的雇员，你可以为此指定下面的查询：

```
select d.*, e.*
from   emp e, dept d
where  e.deptno = d.deptno
order by d.deptno, e.empno
```

这当然会导致一个简单的表格式报告：

Report Window - select d.*, e.* from emp e, dept ...

SQL | Layout | Options

```
select d.*, e.*
from   emp e, dept d
where  e.deptno = d.deptno
order by d.deptno, e.empno
```

Deptno	Dname	Loc	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
10	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	09-06-1981	2450,00	
10	ACCOUNTING	NEW YORK	7839	KING	PRESIDENT		17-11-1981	5000,00	
10	ACCOUNTING	NEW YORK	7934	MILLER	CLERK	7782	23-01-1982	1300,00	
20	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17-12-1980	800,00	
20	RESEARCH	DALLAS	7566	JONES	MANAGER	7839	02-04-1981	2975,00	
20	RESEARCH	DALLAS	7788	SCOTT	ANALYST	7566	09-12-1982	3000,00	
20	RESEARCH	DALLAS	7876	ADAMS	CLERK	7788	12-01-1983	1100,00	
20	RESEARCH	DALLAS	7902	FORD	ANALYST	7566	03-12-1981	3000,00	
30	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00	300,00
30	SALES	CHICAGO	7521	WARD	SALESMAN	7698	22-02-1981	1250,00	500,00
30	SALES	CHICAGO	7654	MARTIN	SALESMAN	7698	28-09-1981	1250,00	1400,00

4:27 14 rows selected in 0,03 seconds

通过在 *LOC* 列指定一个*中断*，你会为你的报告获得更好的结构：

Report Window - select d.*, e.* from emp e, dept ...

SQL | Layout | Options

Auto Update

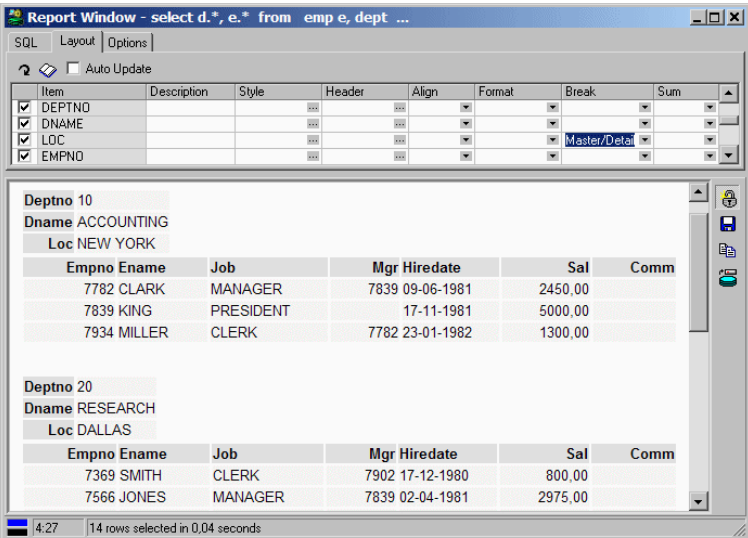
Item	Description	Style	Header	Align	Format	Break	Sum
DEPTNO							
DNAME							
LOC						Break	
EMPNO							

Deptno	Dname	Loc	Empno	Ename	Job	Mgr	Hiredate	Sal	Comm
10	ACCOUNTING	NEW YORK	7782	CLARK	MANAGER	7839	09-06-1981	2450,00	
			7839	KING	PRESIDENT		17-11-1981	5000,00	
			7934	MILLER	CLERK	7782	23-01-1982	1300,00	
20	RESEARCH	DALLAS	7369	SMITH	CLERK	7902	17-12-1980	800,00	
			7566	JONES	MANAGER	7839	02-04-1981	2975,00	
			7788	SCOTT	ANALYST	7566	09-12-1982	3000,00	
			7876	ADAMS	CLERK	7788	12-01-1983	1100,00	
			7902	FORD	ANALYST	7566	03-12-1981	3000,00	
30	SALES	CHICAGO	7499	ALLEN	SALESMAN	7698	20-02-1981	1600,00	300,00

4:27 14 rows selected in 0,03 seconds

一直到（包括） *LOC* 的字段的重复值将被禁止显示。简而言之，每个 dept 记录只被显示一次。在这种情况下，记录被按部门（*DEPTNO* 列）排序是必要的。如果你要重复每个中断后面的标题，请使用 *Break + Header* 选项。

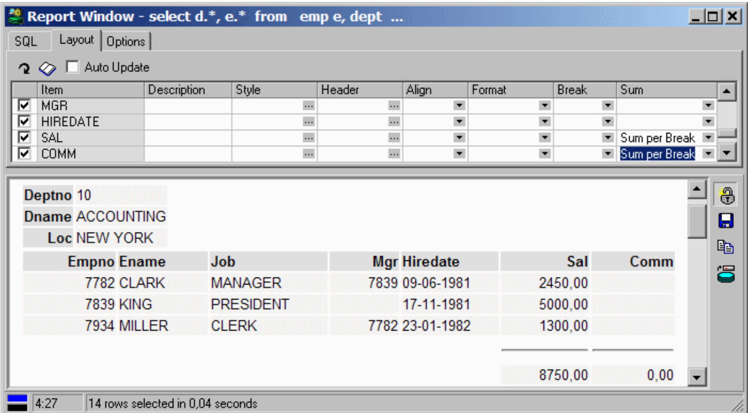
作为一种选择，你可以选择 Master/Detail 版面：



对于一个报告，你可以有多重中断。查询的 order by 子句必须保证结果集在所有的中断列上都被排序。

和

和属性允许你包括一个字段的和：



你可以指定和在中断级、报告级或两者都有被计算。

页眉

要定义页眉，你首先必须设置页的大小（请看第 9.6 章）。报告标题将在每一分页符之后被重复，还可以包括下列变量：

- OSUser – 操作系统用户。
- DBUser – 连接到数据库的用户。
- Database – 用户连接到的数据库。
- Date – 当前日期。
- Time – 当前时间。
- Page – 当前页。

要在页眉里包括这样一个变量，在它之前要有一个 & 字符。例如：

```
All employees (&dbuser@&database)
```

如果你要运行用户为 *scott* 、数据库为 *chicago* 的报告，页眉将象下面这样：

```
All employees (scott@chicago)
```

既然报告被设定为 HTML 格式，你还能把 HTML 代码包括在里面。要包括一个图象，你可以包括 <img> 标签：

```
All employees <img src=logo.gif>
```

要包括左对齐、居中和右对齐信息，你可以使用带有三个 单元格的单行表，并且在相应的单元格里包括这些信息：

```
<table width="100%"> <tr> <td>&dbuser</td> <td align="center">All
Employees</td>  <td align="right">&page</td> </tr> </table>
```

这被显示为：

Scott	All Employees	1

注意，你可以在所有版面项目的描述里使用这些变量和 HTML 代码，但是对于报告标题它们不是最常用的。

9.5 风格库

代替一遍又一遍地定义和重新定义同样的风格，你可以按工具栏里的*风格库*按钮来创建与修改你的标准风格。例如，如果你想把包含 SQL 表达式的所有字段都显示为 Courier New，12 pt，blue ，那么简单地定义一个在库里带有这些属性的 *SQL* 风格，再应用它到相应的地方就可以了。

这里有许多有特定含义的风格名：

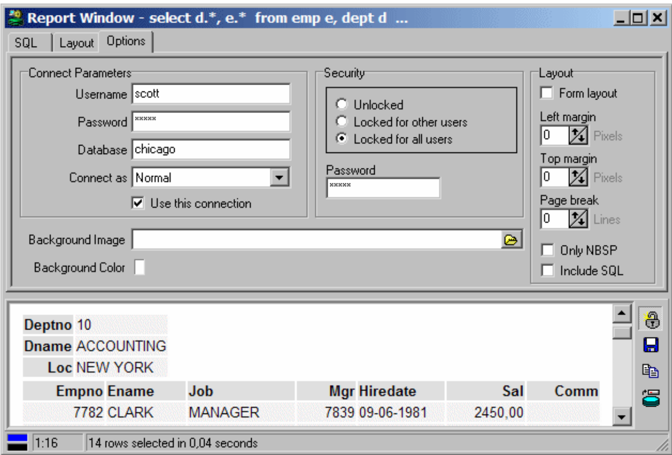
风格名	描述
Default Report	将被应用到报告标题
Default Variables	将被应用到变量
Default Tabular	将被应用到所有表格式表（例如明细记录）
Default Form	将被应用到所有表单式表（例如主记录）
Default Field	将被应用到没有已定义风格的所有字段
Default Field	将被应用到没有已定义风格的所有字段标题

这些是默认的风格，所以在个别的报告里你可以通过定义相应的版面项目风格来越过它们。

注意，所有已用的风格（标准的和定制的）都被存储在报告文件里。当一个报告被运行和一个风格需要被应用时， PL/SQL Developer 将尝试在风格库里定位它。当它不可用时，来自于报告文件的风格将被使用。这将确保若非风格仍然被否决你的报告总是能正确地显示出来。

9.6 选项

你可以为你的报告指定各种不同的选项：



连接参数

连接参数决定了报告查询将怎样运行。例如，如果在 *SYS* 的账号下运行你的报告是必须的，那么你可以在*用户名*的输入框里输入这个名字。另外，你可以输入*口令*，或者你可以保留它为空。如果你保留它为空，当报告被运行时，你将被提示输入口令。如果报告应该总是在同样的数据库实例里被运行，你还可以指定*数据库* 。作为一种选择，你可以在数据库输入框里输入 * 号，它表示报告应该在当前的数据库里被运行。

*使用本连接*选项允许你禁止或允许报告的连接参数。如果它被禁止，当前的 PL/SQL Developer 连接将被使用。

安全

*安全*选项允许你锁定报告定义（SQL ，版面和选项），以至于用户不能查看或改变它。如果连接参数指定了一个有权限的带有口令的账户（例如 SYS），安全选项也许是有用的。如果一个用户能够改变 SQL ，若报告没有被锁定，他或她就能够运行 SYS 帐户下的任何语句。

安全能被设置为下列值之一：

- 不锁定 – 任何人都能改变报告定义
- 锁定其他用户 – 如果你知道报告的*口令*，或者如果你带有与连接参数里被指定的一样的账号在 PL/SQL Developer 里被连接，只有你才能改变报告的定义。
- 锁定所有用户 – 在你能改变报告的定义前，你总是被提示输入报告的*口令*，不管当前你是怎样在 PL/SQL Developer 里被连接的。

如果报告被锁定，报告文件（.rep）就被加密了，以至于不能用文本编辑器来查看或改变它。

版面选项

- 表单版面 – 在默认的情况下，除了主记录以外，所有记录都在表格版面上显示。主记录在表单版面上显示。通过激活这个选项，所有记录都将在表单版面上显示。 如果每一条都有很多 与/或 宽字段，不能显示在一个单一行上，这个选项就可能是有用的。
- 左边距 – 报告左边空白处的像素数量。这个选项仅仅工作于如果你使用 Internet Explorer 作为 HTML 查看器的时候。
- 上边距 – 报告顶部空白处的像素数量。这个选项仅仅工作于如果你使用 Internet Explorer 作为 HTML 查看器的时候。
- 分页符 – 行的数量应在分页符后面显示，报告标题将在每个分页符后面重复再现。
- 仅 NBSP – 当它被允许时，数据里所有的空格都将被转换到 HTML 输出里的 NBSP（不断开空格）。当它被禁止时，正常的空格将被使用。这能影响到版面，因为 HTML 查看器将显示连续的空格为一个空格。

- 包括 SQL – 当它被允许时，报告的 SQL 文本将被包含在 HTML 输出里。

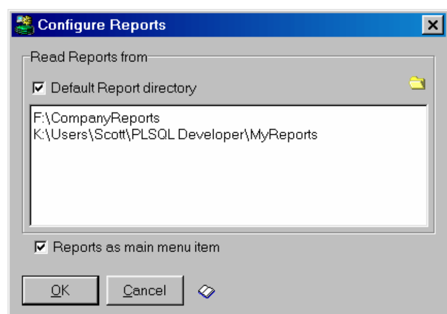
其它选项

- 背景图象 – 应该被显示在这个报告的背景里的图象（.jpg 或 .gif）。你可以使用*选择文件*按钮来查找你的图象文件。
- 背景颜色 – 应该被显示在这个报告的背景里的颜色。

9.7 报告菜单

在默认的情况下，*报告*菜单项目被包括在主菜单里，只包含了标准的报告。这些标准的报告位于 PL/SQL Developer 安装目录的 Reports 子目录里，菜单是这个目录结构的简单表现。选择来自于报告菜单的一个项目将运行那个报告。

你可以在这个目录里添加或移除报告，你还可以通过*工具*菜单里的*配置报告*项目来配置报告菜单。这将调出下列对话框：



*默认报告目录*选项指出了报告菜单是否应该包括来自于 PL/SQL Developer 安装目录的 Reports 子目录，这个目录包含了标准的报告。这个选项下面的编辑区包含了应该被包括的目录列表。你可以用手工键入目录，或者你可以通过按右边的*添加目录到列表*按钮选择一个目录。

注意，报告文件和子目录将被并入到报告菜单里。你应该设法防止在这个目录列表里有重复的子目录和报告文件名，否则，你将在报告菜单里获得不明确的菜单项目。

风格库也可以在这个报告配置对话框里访问到（请看第 9.5 章）。

10. 工程

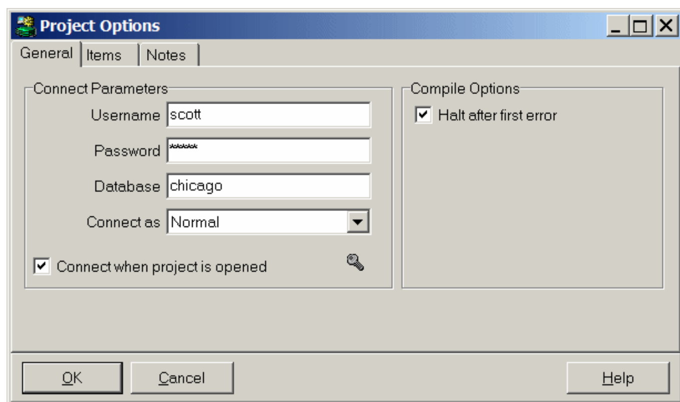
要组织你的工作，你可以使用 PL/SQL Developer 内置的工程概念。一个工程是由源文件、数据库对象、注释和选项的集合构成的，它允许你工作在一组特定项目的范围里，以代替整个数据库或方案。这会使你容易地查找你需要的工程项目，或者容易地从一个地方或数据库移动工程到另外一处。

工程明了你的桌面配置，如果你重新打开一个工程，所有的过去在工程被关闭以前被打开过的项目都将在同样的位置被打开。*自动保存桌面*首选项必须为此激活。这个信息没有被储存在工程定义文件（*工程*.prj）里，而在工程桌面文件（*工程*.dsk）里。

注：要用工程来工作，*使用工程*首选项必须被激活。如果不是这样，主菜单将不包括*工程*项目。对于所有的工程菜单项目，也有相应的工具栏按钮，它们能通过*工具栏*首选项页（请看第 14.9 章）被添加到工具栏里。

10.1 创建一个新工程

要创建一个新工程，请选择来自于*工程*菜单的*新建*项目，这将调入工程选项对话框：



在*一般*页上，你可以定义工程被打开时你要连接到何处，你还可以定义编译错误出现后你是否还要继续编译工程。

*项目*页允许你查看工程项目，这可以是 PL/SQL 程序单元的源文件、测试脚本、SQL 脚本、报告等等。工程的项目还可以是被储存在数据库里的对象，这些对象在源文件没有陈述。当工程被创建时，项目列表显然将是空的。

*注释*页允许你维护工程注释，这些注释能够包含任何东西，例如任务列表、工程决定、设计等等。

10.2 保存工程

设置了选项以后，你可以通过来自于*工程*菜单的*保存*项目来保存工程，这将创建一个带有扩展名 .prj 的工程文件。工程文件包括了项目说明、选项和注释。因此，当你改变选项时或者当你改变注释时，无论你何时添加或移除项目，你都必须保存工程。

保存工程将不会隐含地保存包含在工程里的源文件。

10.3 添加源文件到工程

要添加一个源文件到你的工程，你首先必须在 IDE 里打开源文件。在打开了源文件以后，你可以简单地选择来自于*工程*菜单的*添加到工程*项目。对于程序文件（包含 PL/SQL 程序单元的源），项目在暗中为编译而被激活。这意味着当你生成或建立工程时，这个文件将被包括到编译进程里。其它文件，例如测试脚本、SQL 脚本、报告文件等等，将不会为编译而被激活。

注意，如果可能的话，源文件的路径被储存为相对于工程文件的位置。例如，如果文件被定位于与工程文件相同的目录，路径将不被保存。如果文件在别处，整个路径将被保存。这使工程产生容易地从一个目录移动到另外一个目录的“便携式”效果。

10.4 添加数据库对象到工程

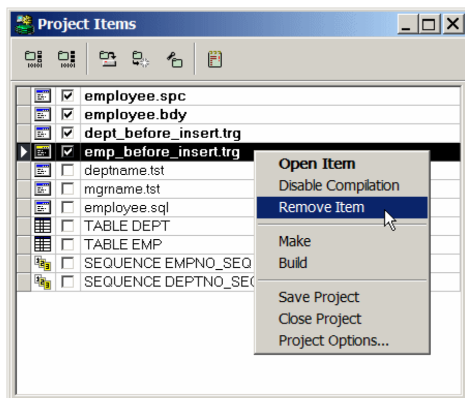
要添加数据库对象到工程，就要在浏览器里右击它，选择来自于弹出式菜单的*添加到工程*项目。正象程序文件一样，PL/SQL 程序单元将在暗中为编译而被激活。其它对象，例如表和序列，则不能被编译。

作为一种选择，你可以从对象浏览器拖放一个数据库对象到工程项目窗口（请看下面）。

如果当前用户拥有数据库对象，则所有者将不被保存到工程文件里。如果另外一个用户拥有这些对象，则所有者将被保存为对象说明的一部分。这使工程从一个用户移动到另一个用户变得很轻松。

10.5 用工程项目工作

在添加了项目到工程以后，你可以选择来自于*工程*菜单的*工程项目*来用工程的项目工作。工程项目窗口可以如下面所示：



最初的四个程序文件允许被编译，两个测试脚本、SQL 脚本以及表和序列不允许被编译。

你可以保持工程项目窗口一直是打开的，以便快速和容易地访问所有工程项目。这个窗口是工程桌面的一部分，当工程被重新打开时它也将在暗中被重新打开。

要编辑一个项目，你简单地双击它就可以。如果项目是一个源文件，它将在一个相应的窗口（程序窗口、测试窗口等等）里被打开。如果项目是一个数据库对象，相应的带有对象定义的编辑器将被打开。

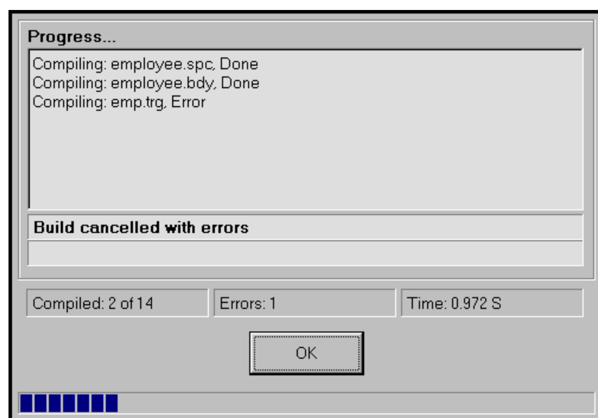
作为一种选择，你可以右击一个项目来调入像上面显示的弹出式菜单，在这里你可以禁止/允许编译项目，或者你可以从工程中移除项目。如果项目是数据库对象，你还可以使用作为子菜单的对象弹出式菜单。

要改变工程项目的顺序，你可以点击行标题把它拖到一个新的位置。这个顺序不仅仅在表面上重要，它还决定了编译的顺序。

10.6 编译工程

要编译一个工程，你可以选择来自于*工程*菜单的*建立*或*生成*项目。

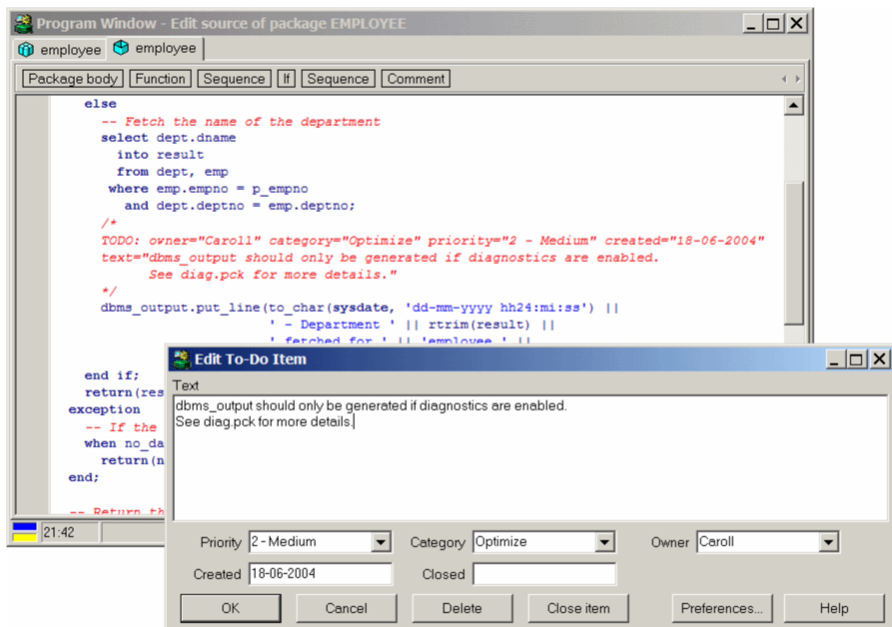
*建立*功能将简单地编译所有的允许编译的工程项目，项目将按显示在工程项目列表里的顺序编译。如果一个项目编译失败，工程选项*出现第一个错误后停止*决定了工程编译是否应该继续进行：



生成功能只编译那些自上次编译以来已经被改变的项目。如果 PL/SQL 程序单元作为一个数据库对象被包括，并且它在数据库里是无效的，那么它也将被重新编译。编译的时间戳被储存在对象桌面文件里。允许编译但是最新的项目，将在编译进程窗口里被列出为*被跳过*。

11. 任务项目

你可以在任何源文件里使用任务项目以产生一个简要的注释，它记载了在这个源文件里需要做的一些事情，你以后可以从任务列表里访问这个信息。这些任务项目被放到了源文件里，作为一种在适当位置的带有特定格式的注释：

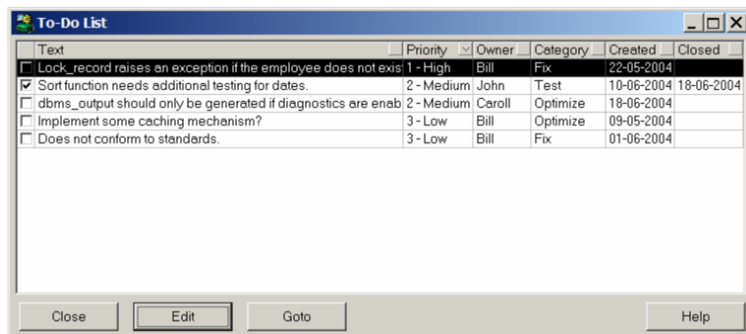


在这个例子里，你可以看到源文件里的注释（开始于文本 *TODO:*）和任务项目编辑器，在这个编辑器里你可以输入文本、优先权、种类、所有者、创建日期和关闭日期。对于优先权、种类和所有者输入框，选择列表的目录能通过首选项（请看第 14.23 章）来定义。

任务项目能通过任务项目编辑器来访问，在这个编辑器里你可以方便地创建、编辑、查看或删除任务项目。只要不改变格式，你还可以直接在源文件里编辑注释。

因为任务项目在源文件里被作为注释来存储，所以它们将始终与源的实际情况一致，不管源是否来自于数据库、来自于文件系统或者来自于一个版本控制系统的旧版本。

要访问指定源文件的所有任务项目，你 可以调用任务列表：

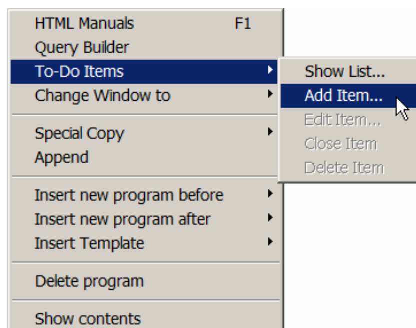


在这里，你可以快速地查找一个任务项目和通过双击它或通过按*转到*按钮在源文件里定位它。

任务列表可以通过点击列标题来排序。

11.1 创建任务项目

要创建一个任务项目，你首先必须把光标放到源文件里适当的位置上。这个位置是注释将要被插入的位置，是接近你要工作的地方，所以这件事是重要的。其次你可以右击这个位置，选择来自于弹出式菜单*任务项目*子菜单的*添加项目*：



在上面提到的任务项目编辑器将显示出来，你可以为新的任务项目输入信息。在按了*确定*按钮之后，注释将被准确地插入到光标位置。

11.2 编辑任务项目

要编辑现有的任务项目，你既可以右击源文件里相应的注释并且选择来自于*任务项目*子菜单的的*编辑项目*，又可以选择*显示列表*项目（如果你当前不接近你要编辑的任务项目的话）。那样你可以从列表里选择任务项目并按*编辑*按钮。

现在，你可以改变任务项目的各种属性并按确定按钮来更新源文件里的注释了。

11.3 关闭任务项目

通过右击源文件里的注释并选择来自于*任务项目*子菜单的*关闭项目*，任务项目就能被关闭。作为一种选择，你可以调用任务项目编辑器并按*关闭*按钮。

11.4 删除任务项目

通过右击源文件里的注释并选择来自于*任务项目*子菜单的*删除项目*，任务项目就能被删除。作为一种选择，你可以调用任务项目编辑器并按*删除*按钮。

你还可以简单地从源文件里删除注释。

12. 窗口、数据库会话和事务

PL/SQL Developer 提供了多会话/多线程环境。

多会话意味着，即使你在 L/SQL Developer 里只登录了一次，各个数据库会话也能被用于测试窗口、SQL 窗口和命令窗口。此外，程序单元编译也将出现在各个数据库会话里。记住，编译是 DDL 语句并在暗中提交当前的事务，如果与其它窗口共享同样的会话，这有可能产生有害的结果。

多线程意味着，测试窗口、SQL 窗口和命令窗口的运行可以同时运行，不会互相妨碍。它的优势是，当长的连续的 SQL 语句正在运行时你可以做其它的工作，你可以中断或关掉会话，你可以测试你的程序单元的锁定行为，等等。

12.1 会话模式

多线程意味着你需要一个多会话的环境。记住，对于任何特定的数据库会话，Oracle 服务器每次只能运行一个语句。如果某些事情运行在不同的会话里，那么他们只能同时运行。所有的开发环境都不允许有不限制数量的会话，所以就需要设置首选项（请看第 14.2 章）以控制 PL/SQL Developer 将要使用的会话的数量。这个首选项被调用到会话模式，可以有下列的值之一：

- 多路会话 – 每一个测试窗口、SQL 窗口和命令窗口都将有它自己的会话，另外的一个会话将被用于编译。这是最灵活的设置，明显地会导致最大数量的数据库会话。另外可能的缺点是，在更新被提交之后，它们只在 X 窗口里可以看到而在 Y 窗口里看不到。
- 双路会话 – 测试窗口、SQL 窗口和命令窗口将共享一个会话，另一个会话将被用于编译。这个模式的缺点是每次只有一个窗口可以运行程序。
- 单路会话 – 所有的窗口和所有的编译都使用同一个会话，这使事务管理变得很困难。在这个模式里调试器被禁止使用。如果你被限制只能使用一个数据库会话，那么你只能使用这个设置了。

12.2 运行于多路会话或双路会话模式

当你在多路会话和双路会话模式里运行测试窗口、SQL 窗口和命令窗口时，工具栏里的*中断*按钮将被激活，这样你可以在任何时间中断运行。中断并不总是成功的，例如在锁定情况下就是这样。这样，如果你有权限，就可以采用会话菜单里的*关掉*项目。

如果你移动当前运行窗口的鼠标光标，它的状态就会改变，以显示它正在运行并且你不能对那个窗口的内容做任何修改。

当运行完成之后，如果事务已经开始，*提交*和*回滚*按钮就会被激活。在多路会话模式里，每一个窗口有它自己的会话，所以也有它自己的事务，*提交*和*回滚*按钮将反映出最上面窗口事务的状态。

13. 浏览对象

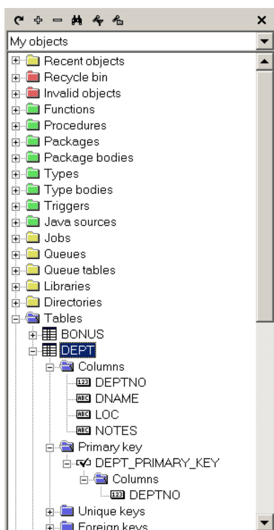
当你在 Oracle 数据库里正在开发一个程序单元的时候，这个程序单元总是会访问其它的数据库对象。记住，如果这不是实际情况，那么这个程序单元根本就没有在这个数据库里的权利。要写这样一个程序单元，你必须确切地知道这些其它数据库对象是怎样被定义的。因此，下面列出的问题对于 Oracle 程序员来说将是很熟悉的：

- 这个列的数据类型是什么？
- 这个表的主键是什么？
- 那个被打包的函数的准确名称是什么？
- 这个参数的默认值是什么？
- 这个触发器的触发条件是什么？
- 哪个对象调用了我正在用的函数？
- 等等...

通过使用浏览器，所有这些信息在 Oracle 词典里都是可利用的，都可以从 PL/SQL Developer 内部访问到。

13.1 使用浏览器

通过选择*工具*菜单里的*浏览器*项目，你能获得可用的浏览器。浏览器显示出了数据库里所有的与开发程序单元有关的基本对象。你可以操纵这个树形视图，这与你使用 Windows 的 Explorer 来操纵文件夹的方法一样。打开*表*文件夹就显示出了可以通过当前会话访问的所有的表，打开一个指定表的文件夹就显示出了与这个表有关的所有属性，打开表下面的 *Foreign keys* 文件夹就显示出了这个表的所有外键，等等。*最近对象*文件夹包括了所有的在 PL/SQL Developer 里最近用过的对象。



无论你何时打开文件夹，从字典里查询的信息都被读进了内存，以使后来的访问尽可能地快。在某些事情被改变后，要强制浏览器重新查询这个信息，你必须选择已修改的对象并按在窗口左上边的*刷新*按钮。你还可以通过选择来自于弹出式菜单的相应项目来刷新对象。要刷新整个浏览器，在按住*刷新*按钮同时请按 *Ctrl* 键。

你可以把对象从对象浏览器拖放到工作空间里，这将调出带有对象定义的编辑器。你可以通过设置首选项来准确地控制什么功能应该被执行（请看第 14.10 章）。

你可以把对象从对象浏览器拖放到编辑器里，这将调出一个弹出式菜单，它的内容依赖于这个对象。你可以选择复制名称（如果必要的话通过所有者加上前缀）、属性、描述和其它有关文本到编辑器里你要放的地方。如果你拖放一个文件夹到一个编辑器里面，

文件夹里所有的对象名都将被复制，这些对象名都将用逗号分隔。你可以设置一个首选项来控制对象名是放在一行上还是多行上（请看第 14.11 章）。

通过在对象图标上按右鼠标按钮，一个前后关系敏感的弹出式菜单就显示出来了，它允许你执行这个对象上的其它功能。这些来自于弹出式菜单的功能在下面的段落里有所描述。

在选择期间，通过使用 *Ctrl* 或 *Shift* 键，你能选择多个对象。如果多个对象被选择，对象的弹出式菜单将只显示那些对所有已选对象可应用的项目，并且拖放功能将同时工作在所有的对象上。

浏览器的首选项（请看第 14.10 章）允许你定义哪些文件夹是可见的、以什么顺序显示。它还允许你定义在*最近对象*文件夹里能看到的对象的最大数量。此外，你还能定义对浏览器对象双击的行为。在默认的情况下，这将扩展相应的节点，你还可以定义这样将调出对象定义编辑器或属性窗口。

创建新对象

要创建一个新的程序单元、表、视图、序列或同义词，请右击这样一个对象或这个对象类型的根目录，选择来自于弹出式菜单的*新建*项目。

使用程序单元的*新建*功能将创建新的程序窗口，初始内容来自于指定的模板，如第 20 章所述。

视图能被创建于 SQL 窗口，初始内容来自于指定的模板，就象程序单元一样。

表、序列和同义词将被显示在指定的编辑器窗口里。表、序列和同义词编辑器在第 8 章里有详细描述。

复制对象

通过选择来自于弹出式菜单的*复制*项目，你可以复制表、序列或同义词。这将调出一个带有已选对象所有属性的编辑器窗口，除了名称以外。如果你要创建一个相似的表来执行某些测试，这就可能是有用的。

复制对象或文件夹名

要复制一个对象名或者一个文件夹里的对象名，请选择*逗号分隔复制*项目，这样做将把名称放到剪贴板上。你可以设置首选项来控制对象名是放在一行上还是放到多行上（请看第 14.11 章）。

查看对象属性

要查看指定对象的属性，请选择来自于弹出式菜单的*属性*项目。你看到的实际属性依赖于对象类型和你的 Oracle 服务器版本。

Properties for EMP_MGR_FK

Property	Value
OWNER	SCOTT
CONSTRAINT_NAME	EMP_MGR_FK
CONSTRAINT_TYPE	R
TABLE_NAME	EMP
R_OWNER	SCOTT
R_CONSTRAINT_NAME	EMP_PK
DELETE_RULE	NO ACTION
STATUS	ENABLED
DEFERRABLE	NOT DEFERRABLE
DEFERRED	IMMEDIATE
VALIDATED	VALIDATED
GENERATED	USER NAME
LAST_CHANGE	12-12-1999 16:12:38

属性被显示在你能放到屏幕上任何地方的模式窗口里。你可以保持属性窗口是可见的（它将停留在其它窗口的顶部）并且继续在另一个窗口里进行编辑。你还可以通过按窗口右上方的按钮来卷起和放开窗口。这样，你能容易地有空间做其它事情，并保持属性是可用的。

查看对象描述

通过选择来自于弹出式菜单的*描述*项目，表、视图、函数、过程、包、类型和方法是能被“描述”的，这在某种程度上类似于我们熟悉的 SQL*Plus 。就象前面提到的属性窗口一样，描述也被显示在一个模式窗口里。

对于表和视图，所有的列和它们相关的属性都被显示在属性窗口里。对于函数、过程和方法，参数和它们的属性会被显示。对于包，它们的函数、过程、变量、类型、常数和异常会被显示。对于类型，方法和属性会被显示。

Columns of EMP

Name	Type	Nullable	Default	Comments
EMPNO	NUMBER(4)			The 4 digit number of the employee
ENAME	VARCHAR2(10)	Y		The last name of the employee
JOB	VARCHAR2(9)	Y	'CLERK'	The job of the employee
MGR	NUMBER(4)	Y		The number of the manager
HIREDATE	DATE	Y	trunc(sysdate)	The date on which the employee was hired
SAL	NUMBER(7,2)	Y		The salary of the employee
COMM	NUMBER(7,2)	Y		The commission of the employee
DEPTNO	NUMBER(2)	Y		The 2 digit number of the department

在描述窗口里，你可以选择行、列或一定范围的单元格。要选择行或列，只点击行标题或列标题再拖动鼠标指针使选择区高亮显示就可以。要选择一个特定范围的单元格，移动鼠标指针越过一个单元格的左边直到光标形态改变，再按鼠标按钮，拖动鼠标直到要选择的区域高亮显示。选择的内容能被复制或打印。要产生一个以逗号分界的已选单元格的拷贝，请右击窗口选择来自于弹出式菜单的相应项目。要导出描述到 CSV 、TSV 、HTML 或 XML 文件，请选择来自于*导出*子菜单的相应项目。要打印出描述窗口的内容，请选择*打印*项目。

通过在描述窗口里双击一个单元格，单元格的内容就会被复制到最上面窗口的编辑器里。

要在描述窗口里对行进行排序，请按你要对行进行排序的那一列的标题按钮。

查看对象定义

所有的对象类型都有定义，要查看这个定义，你可以选择来自于弹出式菜单的*查看*项目。对于所有的程序单元、视图、表、序列和同义词，这个项目都是可利用的。对于表的元素如列或索引，如果你选择了*查看*项目，它将显示出表的定义并导向已选择的元素。

程序单元将显示在程序窗口里，视图将显示在 SQL 窗口里，表、序列和同义词将显示在指定的编辑器窗口里。表、序列和同义词编辑器在第 8 章里有详细描述。

编辑对象定义

通过选择来自于弹出式菜单的*编辑*项目，你可以编辑一个直接来自于数据库的对象定义。你可以通过第 14.2 章描述的首选项来禁止这个特性。

重新命名对象

通过选择来自于弹出式菜单的*重新命名*项目，表、视图、序列和同义词能被重新命名。如果你使用的 Oracle 9.2 或更新版本，你还能重新命名列。带有旧名称的对话框将显示出来，在这里你可以修改和应用它。

删掉对象

有时候删掉一个对象是必要的，它可能已经没有用了，或者必须从一种类型改变为另一种类型（例如从函数改到过程）。要实现这个，只要选择来自于弹出式菜单的*删掉*项目就可以了。

浏览相关对象

大多数对象既有参照又有被参照的对象，通过选择*浏览*项目你就能快速地在浏览器里定位相关的对象。

重新编译对象

对象经常参照数据库里其它对象。无论这些被参照的对象何时改变，参照对象都可能变得无效。在这种情况下，一个红色标志被显示在这个对象的图标里。要编译无效对象，请选择来自于弹出式菜单的*重新编译*项目。如果编译成功，那个标志就不显示了。如果编译失败，你就可以通过选择来自于弹出式菜单的*查看*项目来查看错误。

添加对象源到编辑器

通过选择*添加源到编辑器*项目，程序单元的源文件能被添加到程序窗口或测试 窗口里。

对于程序窗口，源将被添加到当前程序单元后面的一个新页上。这样你可以创建一个包括几个已经存在于数据库里的程序单元的程序文件。

对于测试窗口，在调试会话期间，源将被添加到当前已选择的源的后面。这样你可以在测试窗口里包括源，随后在实际单步进入程序单元以前设置或移除断点。

添加调试信息到对象

在调试会话期间，如果你要查看或设置变量的值，包含这些变量的对象需要包含调试信息。通过选择*添加调试信息*项目，你可以手工添加调试信息。这个菜单项目有一个检验标志，它能显示出对象是否已经包含了调试信息。调试器的首选项部分包含了一个选项，当程序单元被编译时，能自动地添加调试信息，这样你只需要为你没有开发的程序单元使用这个项目。

重新编译参照对象

当你改变一个被其它程序单元或视图参照的对象时，这些参照对象将变为无效。重新编译它们将使它们变为有效或者会发现一些不相容的问题，通过选择*重新编译参照对象*项目，这会很容易地被实现。这将调出会立即被执行的编译无效对象窗口。关于这个工具的更多信息在第 15.3 章能够找到。

测试程序单元

要测试一个程序单元（孤立的函数或过程、被打包的函数或过程、或者对象类型方法），你可以使用测试脚本，其详细资料在第 4 章里有描述。一个测试脚本一般将包含一个简单的函数或过程调用。在运行测试脚本以后，你需要设置输入参数，查看输出参数和函数结果。你可以从浏览器里选择*测试*项目来创建这样一个测试脚本。

查询表和视图

要查看表或视图里的数据，请选择来自于弹出式菜单的*查询数据*项目。对于在浏览器里当前已选的表或视图，一个带有 select 语句的 SQL 窗口就被创建了。如果你把*自动运行 SQL 窗口*首选项设为允许，select 语句将立即被运行。

编辑表数据

如果你要编辑一个表里的数据，当表在浏览器里被选择时，你可以选择*编辑数据*项目。一个带有包括 rowid 的 select 语句的窗口将被创建。如果你把*自动运行 SQL 窗口*首选项设为允许，select 语句将立即被运行。在记录被获取之后，SQL 窗口立即进入编辑模式。

导出表

要导出表的定义和数据，请选择*导出数据*项目，这将为已选表调出*导出表*工具。详细资料请看第 15.4 章。

允许和禁止触发器和约束

通过选择来自于弹出式菜单的*允许*或*禁止*项目，触发器和约束能从对象浏览器内部被允许或禁止。被禁止的触发器和约束被显示为灰色。

运行作业

对于通过 *dbms_job* 包被提交的作业，通过选择来自于弹出式菜单的*运行*项目，你可以明确地运行它。

允许和禁止进入队列和脱离队列

对于队列，通过选择来自于弹出式菜单相应的项目，你可以允许和禁止进入队列和脱离队列操作。在菜单项目前面的检验标志指示了操作在当前是否被允许。

添加对象到自定义文件夹

要添加对象到自定义文件夹（请看第 13.3 章），请选择*添加到文件夹*项目，随后选择一个来自于子菜单的文件夹。

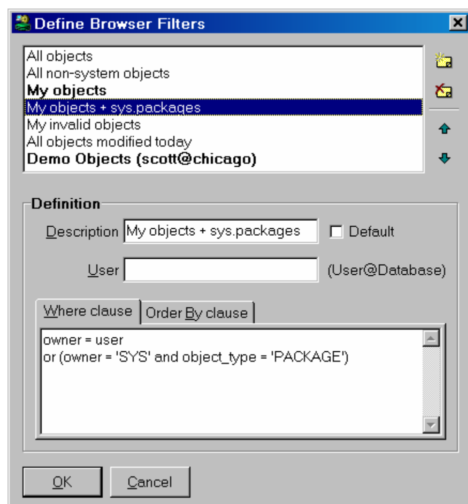
恢复或清理对象

如果你使用的是 Oracle10g 或更新版本，你可以恢复或清理你先前删掉的表，前提是如果这个特性在数据库里被允许的话。对于来自于*回收站*文件夹里的对象，你可以选择*清理*来永远地删除掉表和它的数据，或者你可以选择*恢复*来在原来的名称下或者在一个新的名称下恢复它。

13.2 浏览器过滤器

你可以定义过滤器以决定哪些对象在对象浏览器的系统文件夹里是可见的、这些对象按什么顺序显示。位于浏览器顶部的选择列表允许你快速地在不同的过滤器之间切换。

通过选择*工具*菜单里的*浏览器过滤器*项目或者通过按浏览器上方的*过滤器*按钮，浏览器过滤器就能被定义。你可以在这个对话框里修改、创建和删除过滤器：



过滤器可以被整理出下列属性：

- **描述**
  这个描述在浏览器顶部的选择列表里被显示出来。

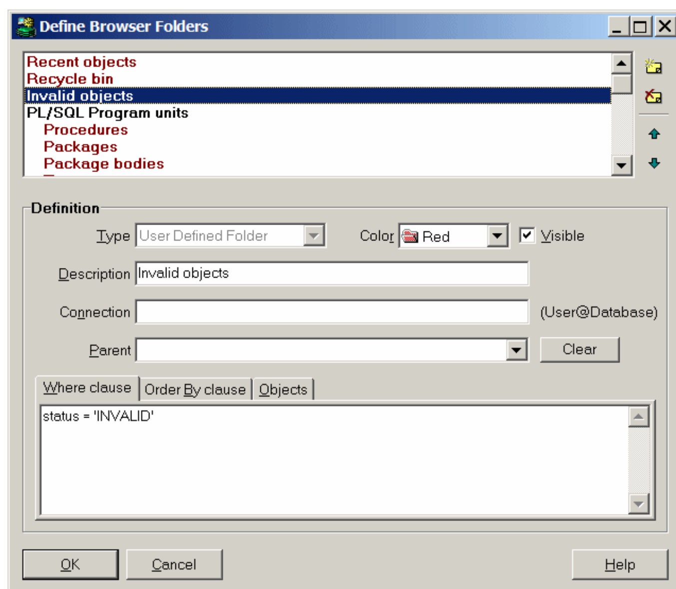
- **用户**
  定义哪一个 Oracle 用户使用这个过滤器是有效的。如果你留此输入框为空白，那么过滤器对所有的 Oracle 用户都是有效的。如果你指定一个用户和数据库（例如 scott@chicago），那么过滤器只在你登录为这个用户和数据库时是可见的。
- **默认**
  如果你检验了这个选项，当你开始运行 PL/SQL Developer 时，已选的过滤器将被激活。你可以为每一个用户都定义一个过滤器，默认的过滤器都以粗体字显示。
- **Where 子句**
  过滤器在 *all_objects* （或 *dba_objects*）视图上定义了一个 select 语句。Where 子句可以用于限制 select 语句的结果集。你可以使用下列的列：
  - **owner – 对象的所有者**
  - **object_name – 对象名称**
  - **object_type – 对象类型（TABLE 、VIEW 、PACKAGE 等等）**
  - **status – 对象状态（VALID 或 INVALID）**
  - **created – 创建对象的日期/时间**
  - **last_ddl_time – 最近被编译对象的日期/时间**
  - **Order by 子句。**
    使用来自于 *all_objects* 视图的列来对浏览器里的对象进行排序。

13.3 浏览器文件夹

对象浏览器的文件夹能用几种方法定制。你可以改变浏览器里文件夹的顺序、每个文件夹的颜色、文件夹的层次，并且你可以基于你当前的连接隐藏指定的文件夹。

你还可以定义你自己的文件夹，并带有对象把它们移民。你既可以明确地添加指定的文件夹到一个目录，又可以定义一个查询把它移民，又可以合并二者。当文件夹在浏览器里被打开时，查询将被执行。

通过选择工具菜单里的*浏览器文件夹*项目或者通过按浏览器上面的*文件夹*按钮，浏览器过滤器就能被定义。你可以在这个对话框里修改、创建和删除文件夹：



在这个对话框的上部，你可以看到所有的文件夹和它们的层次。红颜色指示了系统文件夹，黑颜色指示了自定义的文件夹。一些限制应用到了系统文件夹里。

在文件夹列表的右边，有四个可用的按钮，你可以用它们在当前已选文件夹的位置来添加一个新的文件夹、删除一个文件夹、在列表里上移或下移。对象浏览器里的排序与这个列表里的排序是一样的。

在下部，你可以改变当前已选文件夹的定义：

- **类型** – 说明文件夹的类型：自定义或系统。
- **颜色** – 选择文件夹的颜色，这个文件夹里的对象子文件夹将继承同样的颜色。
- **可见** – 禁止这个选项使对象浏览器的文件夹为不可见，这可以用于隐藏指定的系统文件夹，并且越过*连接*选项。
- **描述** – 对象浏览器里文件夹的描述，不能为系统文件夹而被改变。
- **连接** – 定义哪一个 Oracle 用户的文件夹将被显示出来。如果你保留这个选项为空，则所有 Oracle 用户的文件夹都会被显示。如果你指定一个用户和数据库（例如 scott@chicago），文件夹只在你登录为这个用户和数据库时是可见的。你可以使用 ? 和 * 通配符来匹配多用户。例如，*@prd 为 prd 数据库里的所有用户。

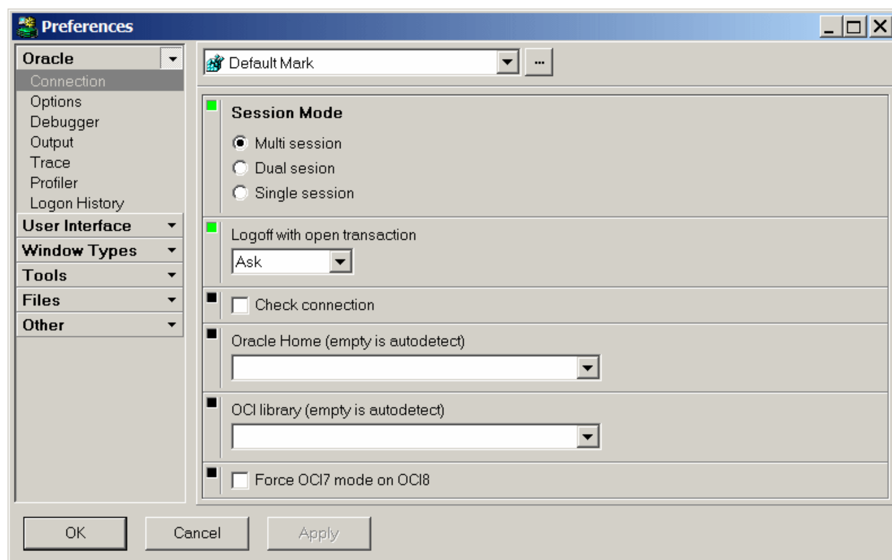
- **父文件夹 – 要按层次放置一个文件夹到另外一个文件夹内部，你可以从列表里选择它的父文件夹。要清除父文件夹，请按*清除*按钮。**
- **Where 子句 – Where 子句可以被用于基于文件夹的查询以限制对象的选择。在上面的例子里，Where 子句被设置为 status = ‘INVALID’，这样就把带有所有无效对象的文件夹的内容移民了。**
- **Order By 子句 – Order by 子句可以被用于基于文件夹的查询以定义已选对象的排序。**
- **对象 – 对于手工建立的文件夹，你可以定义对象列表。你可以按*所有者.名称*格式在每行放置一个对象，例如：*SCOTT.EMP*。注意，通过右击并选择来自于弹出式菜单的*添加到文件夹*，添加一个来自于对象浏览器的单独的对象，这通常是非常方便的。**

Where 子句和 order by 子句可以包括任何 SQL 表达式和来自于 *all_objects* 视图的任何列，这和浏览器过滤器的工作方法一样，请看第 13.2 章以获得更多信息。

注意，过滤器不影响自定义文件夹的对象的内容。

14. 首选项

*首选项*允许你为 PL/SQL Developer 设置不同的首选项。当你选择了这个项目，下列的对话框就显示出来了：



在左边，你可以从六个种类之一中选择一个首选项页（Oracle 、用户界面、窗口类型、工具、文件和其它）。

在顶部，你可以选择一个首选项集，它定义了某一层次的一组参数。在默认的情况下，你可以为你的 Windows 用户（如屏幕截图上方的用户 *Mark*）定义个人的首选项，你还可以为指定的 Oracle 连接定义个人的首选项。无论你何时使用这样的连接来登录数据库，相应的首选项集都会被使用。此外，你的系统管理员还可以在全局层级和 Oracle 连接层级定义默认的首选项。要了解关于首选项集的详细信息，请看第 14.31 章。

在首选项集的下面，你可以看到当前已选的首选项页的首选项（屏幕截图上是 *Oracle – 连接*页）。对于每一个首选项，你将看到当前的值。首选项左边的绿色指示器指示了值在当前首选项集里被设置了。如果指示器是黑色的，则值从不同的集被继承（把鼠标光标放在指示器上面来查看首选项集名）。要恢复首选项到它被继承的值（换句话说，从当前首选项集移除它），你可以点击绿色指示器。

下列章节将详细地描述每一个首选项页。

14.1 Oracle – 连接

- **会话模式**
  控制 PL/SQL Developer 将使用多少个单独的数据库会话。这个特性在第 12 章里有详细描述。

- **注销(有打开的事务)**
  当会话被注销并且这个会话还有打开的事务，默认的行为是提交那个事务。通过这个首选项，你可以定义这样打开的事务被回滚，或者询问你确认信息。

- **检查连接**
  当这个选项被激活时，如果你的数据库连接仍然是活动的，则 PL/SQL Developer 每 60 秒检查一次。如果你的会话被关掉，或者服务器被关闭，PL/SQL Developer 将自动地被注销。

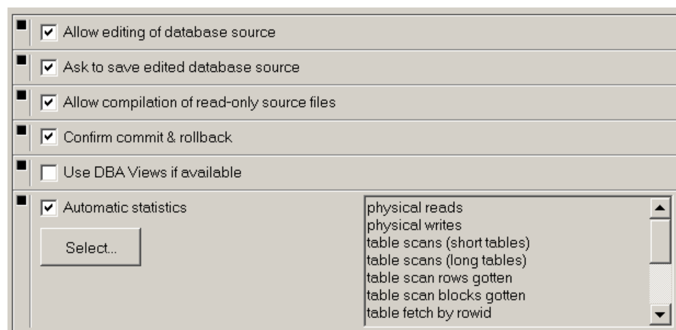
- **Oracle 主目录**
  在默认情况下，PL/SQL Developer 将使用你的电脑里的主 Oracle 主目录。你可以定义带有 Oracle 主目录选择器（一个标准的 Oracle 实用工具）的主 Oracle 主目录。通过选择来自于这个列表里的一个，你可以强制 PL/SQL Developer 使用指定的 Oracle 主目录。

- **OCI 库**
  PL/SQL Developer 将使用被安装的最新版本的 SQL*Net 或 Net8 。如果这个版本的 SQL*Net 或 Net8 引起了问题，你可以强制 PL/SQL Developer 使用另外的版本。DLL 的名称就象 ora72.dll (SQL*Net 2.2)、ora73.dll (SQL*Net 2.3)、oci.dll (Net8 8.1.x) 等等的样子。这些 DLL 被定位于 Oracle 的 bin 目录里。

- **在 OCI8 上强制使用 OCI7**
  一些版本的 Net8 (8.0.3 和 8.0.4) 有一些可能在 PL/SQL Developer 里引起问题的缺陷。配置这个首选项就是用更稳定的 SQL*Net 2 取代

Net8 。Net8 缺陷的例子之一是，当描述过程和函数时，它报告出不正确的参数模式。这个首选项配置的缺点是你不能查询 BLOB 、CLOB 和 BFILE 列，还有被打包的包不能被描述。

14.2 Oracle – 选项

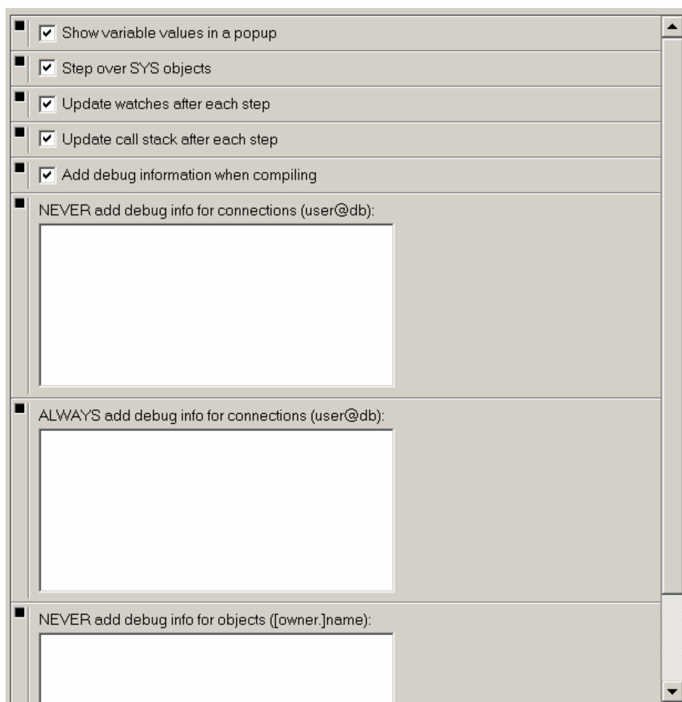


- 允许编辑数据库源
  这个选项控制了程序单元或视图的源是否能从对象浏览器里被直接编辑，或者它需要从文件系统里打开它。
- 询问是否保存刚编辑过的数据库源
  如果你要编辑一个来自于被存储在数据库词典的源里的程序单元（通过右击程序单元并选择来自于弹出式菜单的*编辑*），你可以通过禁止这个选项来防止问你是否保存这些修改到文件里。如果你已经编辑了先前从文件系统里被打开过的程序单元，那么你总是要被问是否保存这些改变。
- 允许编译只读源文件
  如果源文件为只读，当被调入到程序单元窗口里时，它不能被编译。当使用版本控制系统的时候，这个首选项是常用的，它经常把源文件的只读拷贝保留在工作目录里。这样的拷贝不必被编译，因为这会干扰其它对象成员的工作。
- 确认提交和回滚
  无论你何时通过按相应的按钮来提交或回滚一个事务，你都将被要求确认。禁止这个选项将立即提交或回滚，不需要确认。.
- 若可用则使用 DBA 查看
  如果这个首选项被设置，PL/SQL Developer 将试图使用 DBA 视图（*dba_objects* 、 *dba_triggers* 等等）以取代 ALL 视图（*all_objects* 、*all_triggers* 等等）。如果开发者具有某一系统权限，DBA 视图能比 ALL 视图揭示出更多的信息。

- 自动统计
  对于每一个被执行的 SQL 语句和测试脚本，这个选项控制了是否生成统计报

告。如果你禁止了这个选项，统计报告将是空的。通过按*选择…*按钮，一个带有你当前连接的数据库里的所有可用的统计项的列表就打开了。你可以选择你希望包括在统计报告里的哪些统计项。如果你没有访问到动态执行表，*选择…*按钮将被禁止。

14.3 Oracle – 调试器



- **在弹出式菜单里显示变量值**
  如果你禁止了这个首选项，当你在测试窗口里移动鼠标光标越过变量名时，变量值将不会自动地在弹出式菜单里显示。这样，在右击变量名之后，你必须从弹出式菜单里选择设置变量项目。

- **单步跳过 SYS 对象**
  如果你有权查看 SYS 对象的源，如果你禁止了这个选项，调试器就可以单步进入这些程序单元。通常，你不需要这样做，所以这个选项默认为禁止。

- **在每步之后更新监视器**
  如果你禁止了这个选项，通过按测试窗口的调试工具栏里的*更新监视器*按钮，你可以手工刷新监视器。这样做可以在比较慢的配置下使调试的执行快起来。

- **在每步之后更新堆栈**
  如果你禁止了这个选项，通过按测试窗口的调试工具栏里的*更新调用堆栈*按钮，你可以手工刷新调用堆栈。这样做可以在比较慢的配置下使调试的执行快起来。

- **编译时添加调试信息**
  当这个选项被允许时，PL/SQL Developer 里的每一个编译都将自动地添加调试信息，所以在调试会话期间，变量值总是能被查看和设置。如果你禁止了这个选项，你可以从浏览器里手工添加调试信息。

- **从不为连接添加调试信息**
  对于这个连接匹配的列表，调试信息将从不自动被添加，不管“编译时添加调试信息”首选项是否已选。如果你要确信你从不添加调试信息到产品数据库或某一用户，这个功能就可能是有用的。在这里，你可以使用常见的 用户@数据库 语法，也可以使用 * 和 ? 通配符（例如 sys@* 即任何数据库的 sys 用户）。

- **总是为连接添加调试信息**
  对于这个连接匹配的列表，调试信息将总是自动被添加，不管“编译时添加调试信息”首选项是否已选。如果你要确信你总是添加调试信息到开发数据库或某一用户，则这个功能可能是有用的。在这里，你可以使用常见的 用户@数据库 语法，可以使用 * 和 ? 通配符（例如 *@devdb 即 devdb 数据库的所有用户）。

- **从不为对象添加调试信息**
  对于这个对象列表，调试信息将从不被添加。当在特定的情形下添加调试信息时，一些版本的 Oracle 服务器能引起错误。你将在编译期间看到具有代表性的 ORA-00600 、ORA-03113 或 PLS-00801 错误。如果这些问题出现了，你可以添加相应的对象到这个列表（所有者.名称）里，直到 Oracle 提供了修补。

14.4 Oracle – 输出

- [x] Enabled
- [x] Clear before execute

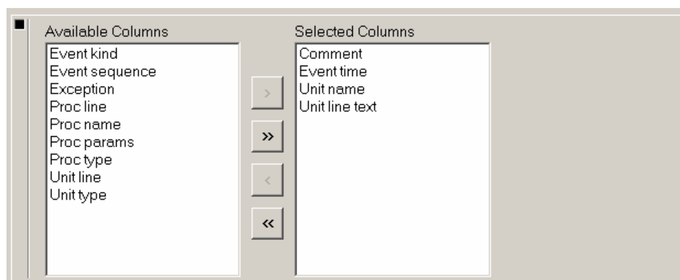
Buffer size
10000

- **允许**
  在 SQL 窗口和测试窗口里，在运行了脚本之后，调用 *dbms_output.put_line* 将被缓冲和显示。在每个单独的 SQL 窗口和测试窗口的输出页上，这个首选项可以被否决。

- **执行前清除**
  在运行 SQL 窗口或测试窗口之前，当前的输出页将被清除。这样，输出页将总是只包含最近运行的输出。

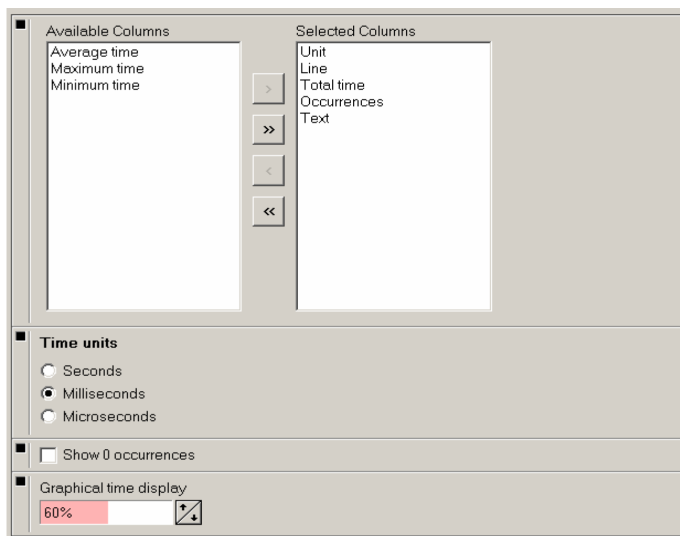
- **缓冲区大小**
  Dbms_output 缓冲区字节的大小。在一个运行期间如果多于这个数的字节被输出，异常将会出现。

14.5 Oracle – 跟踪



这些首选项控制了测试窗口的*跟踪*页（请看第 4.11 章）。你可以控制来自于跟踪表的哪一列应该被显示、应该怎样排序。

14.6 Oracle – 概览图

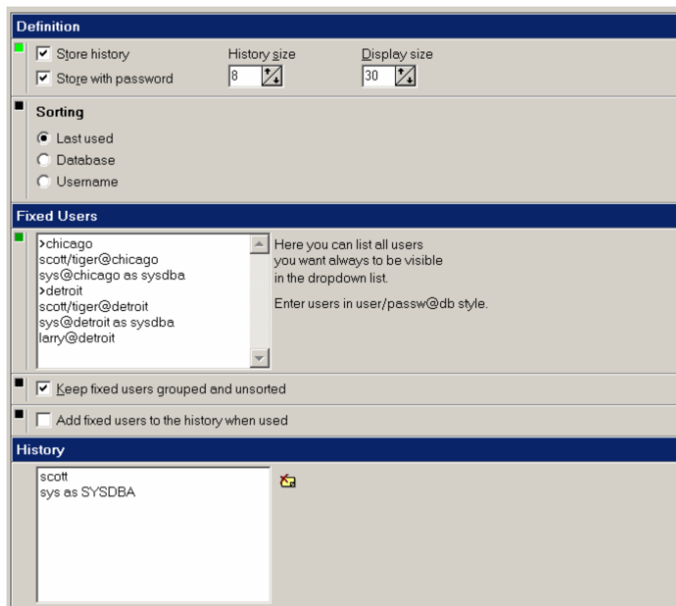


这些首选项控制了测试窗口的*概览图*页（请看第 5.3 章）。

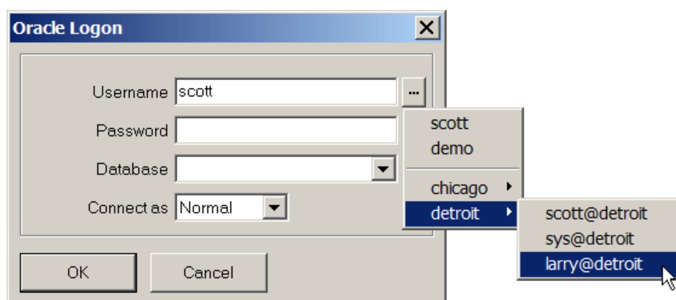
- **列**
  控制来自于概览图表的哪些列应该被显示、怎样排序。
- **时间单位**
  定义了合计时间、最大时间、最小时间和平均时间列的单位。
- **显示 0 具体值**
  确定了被报告的源行是否被显示，这个源行带有执行了 0 次语句获取的具体值。

- **图形化时间显示**
  合计时间列将显示出一个相关时间的图形化的表述。你可以控制这个条的颜色的亮度，或者使它不可见。

14.7 Oracle – 登录历史



如果你激活了*存储历史*首选项，PL/SQL Developer 将在登录历史里保存用户名、口令和数据库信息。当你以后重新开始运行 PL/SQL Developer 时，登录对话框允许你快速地选择以前用过的账号：



为了安全起见，你可以不要求在历史里储存口令，即使 PL/SQL Developer 把它们加密也要这样做。这意味着，如果某人要在你的电脑里访问 PL/SQL Developer ，他们就不能使用你的账号连接到数据库。在这样的情况下，你应该禁止*带口令存储*选项。*历史大小*和*显示大小*首选项控制了多少个连接将被储存（最早的一个将从历史里被移除）、多少个将被显示在屏幕上。*排序*选项控制了在选择列表里连接应该怎样排序。

注意，即使你已经在登录历史里选择了储存口令，你仍然可以通过在登录期间按*回车*键或*确定*按钮时按 *Ctrl* 键来预防这件事。这样你可以防止有权限账号（例如 SYS 或 SYSTEM）的口令能被重新调用。

你还可以在*固定用*户部分定义许多固定的账号，可以带有口令也可以不带有口令。这些固定账号将始终被显示并且永远不会自动被移除。你还可以为这些固定用户通过添加 *>menuname* 来定义子菜单项目。在上面的例子里，创建了 *chicago* 和 *detroit* 的子菜单：

```
>chicago
scott/tiger@chicago
sys@chicago as sysdba
>detroit
scott/tiger@detroit
sys@detroit as sysdba
larry@detroit
```

在这种情况下，子菜单为数据库而被创建，但是你可以创建任何种你喜欢的种类。例如，*开发*、*测试*、*产品*。

如果*保持被组合的未排序的固定用户*选项被激活，固定用户将被显示在来自于历史的一个分开的部分里，并且不会被排序。如果*固定用户一经使用就被添加到历史*选项被激活，在登录期间选择固定用户也将添加到历史。

在*历史*部分里，你可以看到在 PL/SQL Developer 里哪些用户作为登录结果被添加到了历史。要删除指定的项目，请选择它并且按在列表旁边的*删除项目*按钮。

14.8 用户界面 – 选项



- 自动保存用户名
  这个选项可以是已检验，用来自动地保存每一个 PL/SQL Developer 会话的用户名。当你以后重新启动 PL/SQL Developer 时，最近使用的用户名将被默认在登录对话框里。

- 自动保存桌面
  当这个选项被激活时，如果你重新启动 PL/SQL Developer ，所有在 PL/SQL Developer 停止运行时被打开过的文件都会自动地被重新打开。当前打开的不与文件关联的命令窗口或解释计划窗口也将被恢复。此外，主窗口和对象浏览器的大小也将被恢复。

- 使用内置 HTML 阅览器
  在默认的情况下，PL/SQL Developer 将使用内置的 HTML 查看器来显示 HTML 帮助文档。如果你有 Microsoft Internet Explorer 4 或更新版本，你可以禁止这个选项而使用 Microsoft 的 HTML 组件以代之。这将在弹出式菜单里给你一些额外的特性，并且将继承产生于 Internet Explorer 的首选项。

- HTML 帮助窗口:"固顶"
  如果这个选项被允许，HTML 帮助窗口将停留在顶部，直到你关闭它或者把它最小化。如果它被禁止，HTML 帮助窗口将作为一个单独的任务显示出来，它在 Windows 任务条里有自己的按钮。这样，无论你何时需要它，你都可以把 PL/SQL Developer 主窗口或帮助窗口带到前面。

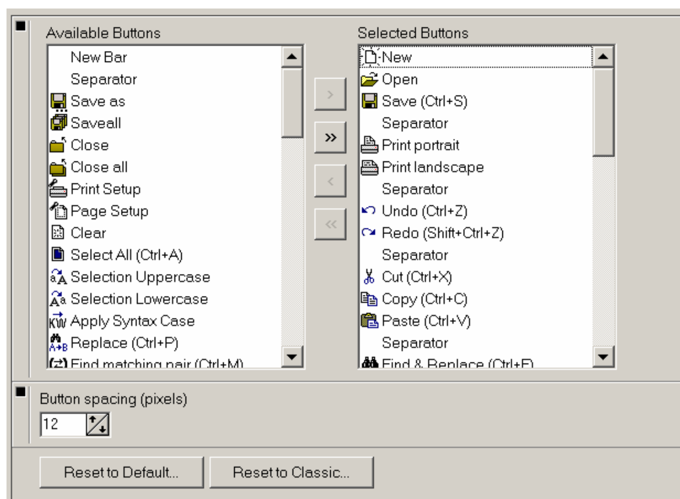
- 使用多行制表符
  如果一个窗口包含了多个标签页，这个首选项控制了万一出现溢出时这些标签页是否可以被分为多个行。如果这个选项被禁止，万一出现溢出情况两个卷动按钮将显示出来。这当前只应用于程序窗口。

- 在窗口标题里显示完整的文件路径
  在默认的情况下，只有文件名才被显示在窗口标题里。如果这个选项被允许，完整的路径将被显示。

- 使用工程
  如果这个选项被允许，只有*工程*主菜单项目才是可见的。如果你没有正在使用的

工程，就禁止它，这样它不会在主菜单里占用任何空间。要了解关于工程的更多资料，请看第 10 章。

14.9 用户界面 – 工具栏



你可以控制哪些按钮应该是可见的，以什么顺序显示。使用*分隔符*项目来分隔相关按钮的组。*新工具栏*按钮可以被用于在先前的工具栏下面创建新的工具栏。

*复位到默认*按钮将复位工具栏到默认的状态，*复位到典型*按钮将复位工具栏到 PL/SQL Developer 3 的版面，使用*按钮间距*值控制了每一个之间的间距。

14.10 用户界面 – 浏览器

Maximum recent objects
10
Double-click action
Expand / Collapse
Drag & Drop action
View
Show datatypes
Smooth refresh
Confirm recompile
Filters...
Folders...

这些首选项控制了对象浏览器的外观和行为（请看第 13 章）。

- 最近对象最大数量
  控制在*最近对象*文件夹里多少个对象是可见的。当多于这个数字的对象将要被显示时，最近最少用的对象将被移除。
- 双击行为
  控制了当你双击一个浏览器里的对象时哪种行为将被执行。默认的行为是展开或折叠项目，但是你还可以定义窗口被打开以查看或编辑对象定义或查看它的属性。你还可以定义在包或类型环境里说明与体两者都被查看或编辑。
- 拖放行为
  控制了当你拖放一个对象到工作空间时哪种行为将被执行。默认的行为是查看项目，但是你还可以定义窗口被打开以查看或编辑对象定义或查看它的属性。你还可以定义在包或类型环境里说明与体两者都被查看或编辑。
- 显示数据类型
  当这个选项被允许时，列的数据类型（表或视图）、属性（对象）和参数（函数和过程）都将被显示在浏览器里。
- 平滑刷新
  在一些系统上，这个选项可以使浏览器刷新操作慢下来（例如改变浏览器过滤器、登录为不同的用户等等）。如果这是你应该禁止的情况，当浏览器被刷新时，这可能就会导致一些屏幕闪烁。
- 确认重新编译
  当这个首选项被允许时，在浏览器里重新编译程序单元将以成功或失败信息为结果。当它被禁止时，你将只能在编译失败时看到信息。

过滤器（请看第 13.2 章）按钮和文件夹（请看第 13.3 章）按钮将调用相应的对话框。

14.11 用户界面 – 编辑器

Indent
Automatic Step (chars) 2
Tabs & Wrapping
Smart tab Tab size 2
Smart fill
Use tab character
Wrap lines Visible margin 0
Allow cursor after end of line
Syntax Highlighting
Enabled

	Bold	Italic	Color Text	Bkg
Keywords	✓			
Comment		✓		
Strings				
Numbers				
Symbols				
Custom	✓			

Keyword case
Other
Display line numbers
Highlight edit Line
Wrap to start / end of file when text is not found
Highlight parentheses
Paste comma separated values on one line

缩进

- **自动**
  确定在按回车键以后编辑器是否将自动缩进。
- **步长（字符）**
  当缩进或撤消缩进一个选择之后编辑器将缩进的字符数。

跳格与换行

- **智能跳格**
  将导致编辑器跳格到先前行相关的位置。
- **跳格大小**
  确定一个跳格将使用的字符数。

- **智能填充**
  如果可能编辑器将用跳格替换空格。
- **使用跳格字符**
  当被禁止时，只有空格才能用于跳格和缩进。
- **自动换行**
  比窗口宽度还宽的行将在文本行上被延续。
- **允许将光标放到行末端的后面**
  当被允许时，你可以将光标放到行末端的后面。当你在这里插入文本时，行将首先用空格来填补。
- **可见边线**
  将在被指示的位置显示垂直线。 0 = 禁止。

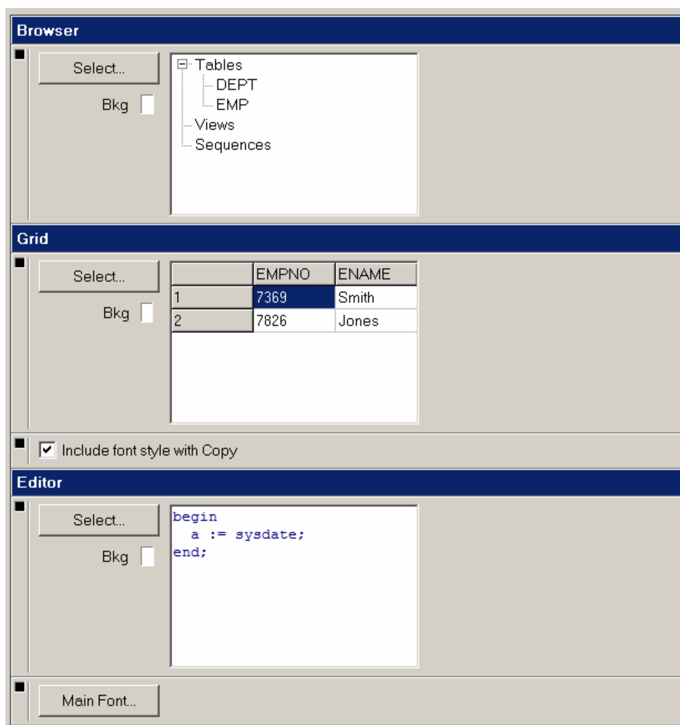
语法高亮

这组首选项控制了用于编辑器里的语法高亮。你可以允许或禁止语法高亮，并且定义关键词、注释、字符串、数字和符号（:= 、 => 、|| 、等等）的字体和颜色。对于关键词，你还可以控制它们是否应该被转换到大写、小写或以大写字母写。你还可以定义定制的语法高亮、定义应该被高亮的词。按*定制关键词*按钮将调出允许你定义这些词的编辑器。

其它

- **显示行数**
  这个选项为已检验会使行数在程序编辑器和测试窗口的左边空白处变为可见。
- **加亮编辑行**
  控制了文本光标所在位置的行是否高亮显示。
- **文本没有被发现时自动转到文件的开始/末尾**
  当使用查找功能时，如果要搜索的文本没有找到，搜索将在文件的开始处（向前搜索时）或文件的末尾处（向后搜索时）继续进行。
- **圆括号高亮**
  当被允许时，如果光标在打开或关闭圆括号处，编辑器将加亮圆括号对。
- **粘贴逗号分隔项目于一行上**
  控制了在对象浏览器里或在描述窗口里被选择的多个项目被粘贴到编辑器里的一行上还是多行上。

14.12 用户界面 – 字体



字体的首选项控制了浏览器字体、编辑器字体和用于各种诸如 SQL 窗口、属性窗口等等的表格的字体。通过按*选择*按钮，你可以改变字体的名称、风格和颜色。你还可以为所有的对话框设置主字体。

14.13 用户界面 – 代码助手

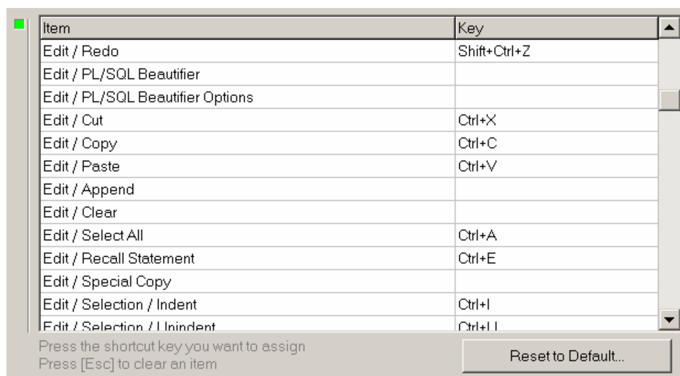
Automatically activated
Delay (ms)
500
Coding Style
Smart
Use original case if possible
Describe Users
Tables
Views
Sequences
Functions
Procedures
Packages
Types
Describe Context
Minimum characters
3
Describe standard functions (e.g. to_char)

当你键入数据库对象的名字时，代码助手将自动地显示关于它们的信息（请看第 16.4 章）。这个首选项页允许你定义这个特性的行为。

- 自动激活
  在某一个延迟之后，代码助手能自动地被调用（请看下面）。你还可以通过功能键选择手工激活代码助手。
- 延迟
  编辑器在显示代码助手列表之前将等待的毫秒数。
- 代码风格
  控制了当你选择了已选的项目时它们将怎样被插入到编辑器里：
  - Smart – 代码助手将考虑被描述的对象来决定风格。
  - Init Caps – 每个词（用下划线分隔）的首字符大写。
  - Lowercase – 所有字符都转换到小写。
  - Uppercase –所有字符都转换到大写。
- 如果可能使用原来的大小写
  这个选项被允许时，如果可能的话，代码助手将确定来自于存储于 Oracle 词典的源里的标识符的大小写。这将应用到所有的程序单元和它们的元素（参数、类型等等）以及应用到查看列，并且越过了上面描述的代码风格首选项。如果原始的大小写不能被确定，代码风格将被应用。
  你可以因执行的原因要求禁止这个特性。

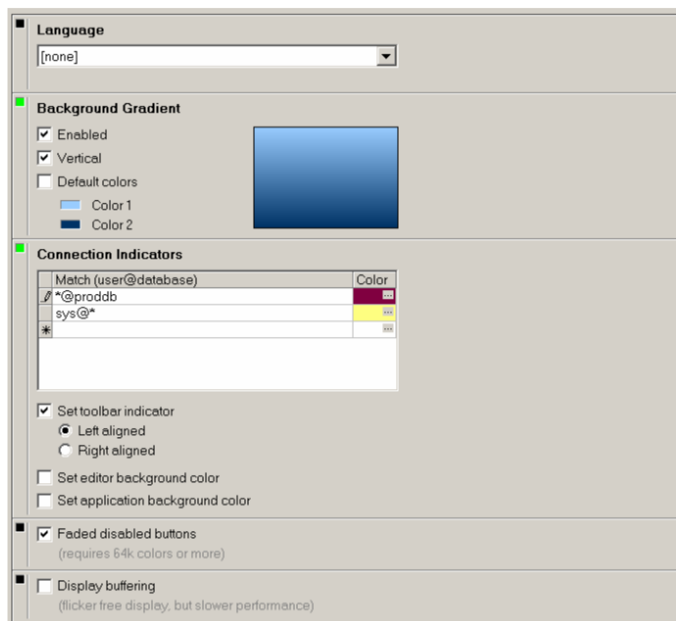
- **描述用户**
  确定了当你键入一个后面跟随着句点的用户名时被用户拥有的对象是否被列出来。如果这个选项被允许，你还可以定义哪些对象类型你要包括在里表里。
- **描述前后关系**
  确定了代码助手是否应该描述当前用户、编辑器和程序单元的前后关系。
- **最少字符数**
  确定了在前后关系描述能自动地被调用之前有多少个字符的词需要被键入。注意，你始终可以手工调用代码助手，即使字符数没有被键入也是这样。
- **描述标准函数**
  在默认的情况下，代码助手将描述标准的函数诸如 to_char 、add_months 等等。如果你很熟悉这些函数，你可以禁止这个选项。

14.14 用户界面 – 键配置



在这一页，你可以为 PL/SQL Developer 里的所有功能定义你自己的键配置。仅仅选择列表里的功能再按你要用于这个功能的组合键就可以。要清除指定功能的热键，请从列表里选择它并按 *ESC* 键。

14.15 用户界面 – 外观



- **语言**
  如果你有多国语言版本的 PL/SQL Developer ，你可以在这里选择一种语言。随后，所有的说明和信息都将显示在那种语言里。
- **背景梯度**
  允许你定义 PL/SQL Developer 工作空间的颜色和梯度。如果你不要求使用梯度，只用单一的颜色，请把颜色 1 和颜色 2 设为同样的值。
- **连接指示器**
  当 PL/SQL Developer 正在使用指定的连接时，这些选项允许你提供一些可见的指示。你可以以用户/数据库格式定义许多连接匹配并与颜色关联起来。随后，这个颜色能与左或右对齐的工具栏指示器、编辑器背景或应用程序背景关联起来（越过背景梯度）。
  连接匹配可以包含 * 和 ? 通配符。在上面的例子里，当以任何用户连接到 *PRODDB* 数据库时，编辑器背景和右对齐工具栏指示器将是红色的；当以 *SYS* 用户连接到任何数据库时，它将是黄色的。列表里的第一个项目将具有优先权，所以当以 *SYS* 用户连接到 *PRODDB* 数据库时指示器将是红色的。
- **淡色禁止按钮**
  当被允许时，被禁止的按钮将以淡颜色显示。当被禁止时，标准的灰色外观被用于被禁止的按钮。对于 256 色显示，被禁止的按钮总是灰色的。

- **显示缓冲**
  当移动窗口或重新确定窗口大小时，屏幕将显示出比较少的闪烁，但显示执行可能稍微慢一些。

14.16 用户界面 – 日期/时间

Date
dd-MM-yyyy
User defined
Windows format
Oracle format
Time
hh:mm:ss
User defined
Windows format

在这一页上，你可以定义将被用于 PL/SQL Developer 的日期和时间格式。

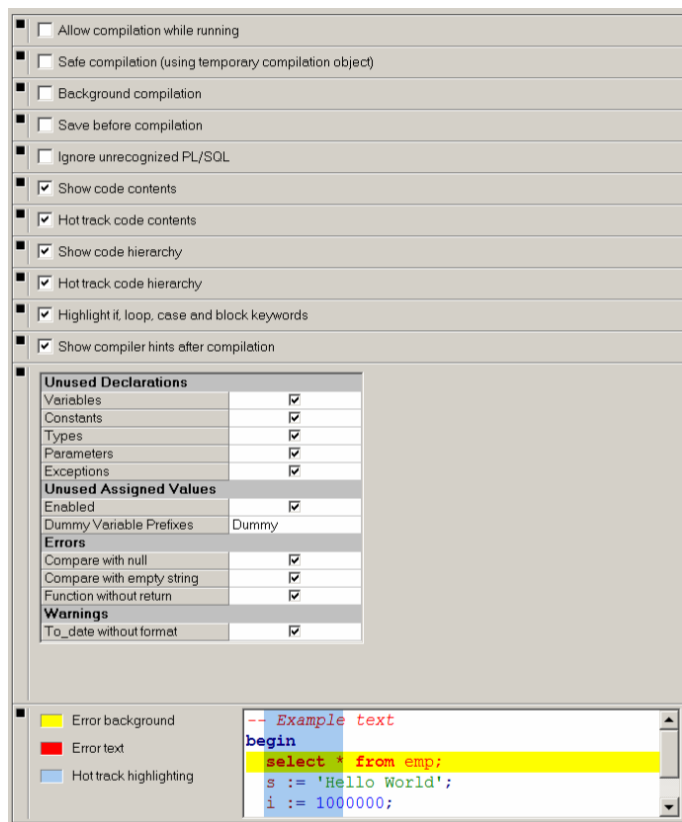
日期格式可以用三种方法定义：

- **自定义**
  一种被明确定义的日期格式，遵循与 Windows 地区设置一样的规则。
- **Windows 格式**
  与 Windows 地区设置一样的格式。
- **Oracle 格式**
  与你的 Oracle 环境（NLS_DATE_FORMAT）定义的格式一样的格式。

时间格式可以有两种方法被定义：

- **自定义**
  一种被明确定义的时间格式，遵循与 Windows 地区设置一样的规则。
- **Windows 格式**
  与 Windows 地区设置一样的格式。

14.17 窗口类型 – 程序窗口



这些首选项影响了程序窗口的行为（请看第 3 章）。

- **允许运行时编译**
  在多路或双路会话模式里，当你正在运行测试脚本或 SQL 脚本时，你在技术上能够编译程序单元，这可能导致锁定。使用这个首选项，这种情况可以被预防。
- **安全编译(使用临时编译对象)**
  当被允许时，每一个编译都首先通过使用临时编译对象来测试。当这个测试编译成功后，实际的对象就被编译了。在一个无效对象可能引起问题的环境里，如果你正在编译对象，这个选项就可能是有用的。
- **后台编译**
  当被允许时，编译在后台线程里使用单独的会话被执行。这样，当编译正在发生时，你可以继续进行其它工作。这将为每一个程序窗口导致额外的数据库会话。

注意，这个首选项仅仅有效于会话模式首选项（请看第 14.1 章）被设为多路会话的情况下。

- 编译前保存
  如果你要保证数据库对象与文件系统是同步的，你可以允许这个选项。
- 忽略未被承认的 PL/SQL
  当被允许时，程序窗口将忽略所有的不以“create or replace <程序单元类型>”开始的 SQL 。程序文件里其它的 SQL 或 SQL*Plus 命令将被忽略。
- 显示代码目录
  这个首选项控制了代码目录窗格是否将被显示，要了解更多详细资料请看第 16.11 章。
- 热跟踪代码目录
  控制当鼠标在代码目录窗格里的项目上移动时编辑器里的 PL/SQL 码是否将自动地被高亮显示。
- 显示代码层次
  这个首选项控制了代码层次窗格是否将被显示，要了解更多详细资料请看第 16.12 章。
- 热跟踪代码层次
  控制当鼠标在代码层次窗格里的项目上移动时编辑器里的 PL/SQL 码是否将自动地被高亮显示。
- If，loop，case 和块关键词高亮显示
  那时光标将被定位于 if / then / else / elsif /end if 关键词、loop / end loop 关键词、for / loop / end loop 关键词、while / loop / end loop 关键词、case / when / then / else / end case 和 declare / begin / end 关键词。
- 编译后显示编译器提示
  编译器提示提供了关于未用声明、未用分配、潜在错误和警告的信息。当这个首选项被允许时，这些提示将自动地被显示在编译的后面、在编译错误的下面（如果有的话）。如果你禁止了这个首选项，你必须明确地选择来自于*工具*菜单或工具栏里的*显示编译器提示*项目。
- 提示列表
  你可以允许或禁止每一个单独的提示，适当的时候控制提示的其它属性。每一个提示和属性的解释都显示在提示列表的下面。
- 错误背景
  选择编译错误文本的背景的颜色。
- 错误文本
  选择编译错误文本的颜色。

- **热跟踪高亮显示**
  为代码目录热跟踪、代码层次热跟踪以及 if 、loop 、case 和块关键词选择高亮文本的背景颜色。

14.18 窗口类型 – SQL 窗口

- [ ] AutoCommit SQL
- [ ] AutoCommit posted records
- [x] AutoExecute queries
- [ ] AutoSelect statement
- [x] Ask to save new windows
- [ ] Linked query in new window
- [ ] Number fields to_char
- [ ] Date fields to_char
- [x] Show dictionary info in result grid

Null value cell color

Records per Page

- (•) Automatically determined
- ( ) Fixed
  100
- ( ) All records

Query By Example

- [ ] Case insensitive
- [ ] Automatic contains
- [ ] Ignore time fraction

Number Layout

- ( ) Left aligned
- ( ) Right aligned
- (•) Formatted
- ( ) Formatted with thousand separator

这些首选项影响了 SQL 窗口的行为（请看第 6 章）。

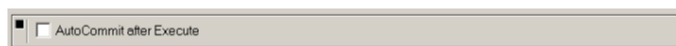
- **自动提交 SQL**
  这个选项控制了在 SQL 窗口里执行的 DML 语句是否被自动提交。
- **自动提交已录入的记录**
  这个选项控制了通过结果表格录入的记录是否被自动提交。
- **自动执行查询**
  如果这个选项被允许，在为表或视图对象选择*查询数据*或*编辑数据*时被创建的 SQL 窗口将立即运行。如果这个选项被禁止，你可以在执行前修改已产生的 select 语句。
- **自动选择语句**
  如果你在 SQL 编辑器里有不只一个 SQL 语句（通过分号分隔），SQL 窗口在运行前将自动地在光标当前被定位的地方选择语句。这样，你不必明确地选择你自己的语句。

- **询问保存新窗口**
  当你关闭刚被创建的 SQL 窗口时，你通常都会被问是否要保存它到文件里。如果你禁止了这个选项，这个确认将被忽略。如果你已经修改了一个先前打开的 SQL 文件，你将总是被问是否保存这些改变，不管这个首选项是否已选。
- **在新窗口里执行连接查询**
  当这个首选项被设置时，执行连接查询（请看第 6.4 章）将创建一个新的 SQL 窗口。如果没有设置，连接查询将在当前的 SQL 窗口里被执行。
- **数字字段 to_char**
  当这个首选项被允许时，数字字段在服务器里将被转换为字符值，会话的 NLS 参数将决定值的格式，L/SQL Developer 将不另外执行数字格式化或确认。
- **日期字段 to_char**
  当这个首选项被允许时，日期字段在服务器里将被转换为字符值，会话的 NLS 参数将决定值的格式，PL/SQL Developer 将不执行任何另外的日期格式化或确认，并且在编辑日期字段时日期采集者将不可用。
- **在结果表格里显示字典信息**
  当被允许时，下列词典信息将被显示在结果表格里：
  - **列数据类型、可选项和注释都将被显示在状态行上。**
  - **对于带有检查指定值的检查约束的列（例如 col in（值 1，值 2，...）），查找列表也将显示出来。**
  - **对于带有外键约束的列（针对少于 1000 行的小表），查找列表也将显示出来。**
- **Null 值单元格颜色**
  这个颜色被用于显示 null 值，这使 null 值与空格值、不直接显示在表格里的值（例如 LOB 和 Long）很容易被辨认出来。
- **每页记录数**
  这个首选项决定了 select 语句最初要获取多少条记录、每次按 *下一页*按钮在 SQL 窗口的结果表格里显示多少条记录。
  - **自动确定 – 结果表格的大小决定了多少条记录被获取。**
  - **固定 – 获取指定的记录数。**
  - **所有记录 – 当查询被执行时，所有记录立即都被获取。**
- **数字版面**
  这个首选项控制了怎样显示数字列：
  - **左对齐 – 左对齐显示值，不带任何格式。**
  - **右对齐 – 右对齐显示值，不带任何格式。**
  - **指定格式 – 右对齐显示值，带有符合字段的精度和范围的格式。**

    - 用许多分隔符指定格式 – 类似于前面的选项，但还将包括千位分隔符（例如 1,277.65）。

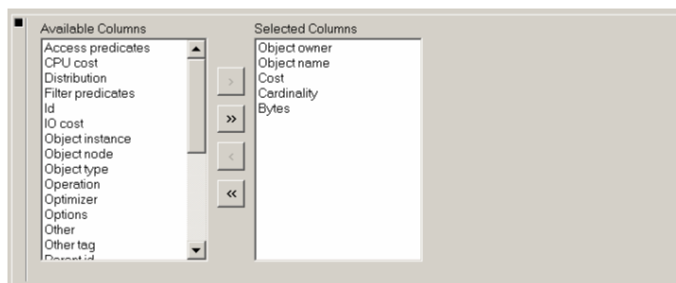
- 实例查询
  这些首选项控制了在 QBE 模式期间 SQL 窗口的默认行为（请看第 6.3 章）：
    - 不区分大小写 – 字符字段的查询值被视为不区分大小写。
    - 自动包含 – 字符字段仅仅需要包含查询值。
    - 忽略日期字段时间部分 – 如果查询值不包含时间部分，日期字段的时间部分将被忽略。

14.19 窗口类型 – 测试窗口



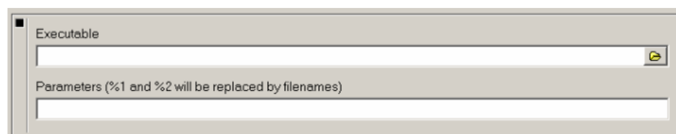
- 在执行后自动提交
  这个选项控制了被执行的测试脚本是否自动地被提交。

14.20 窗口类型 – 计划窗口



这些首选项控制了计划表的哪一个列应该被显示在计划窗口里（请看第 5.1 章）、它们应该按什么顺序显示出来。

14.21 工具 – 差异



在这个首选项页上，对于比较用户对象工具，你可以选择外部的差异工具（请看第 15.9 章）。

14.22 工具 – 数据生成器

General Preferences

Date Format

Commit every...

0

Delay (mSec)

0

Number of items

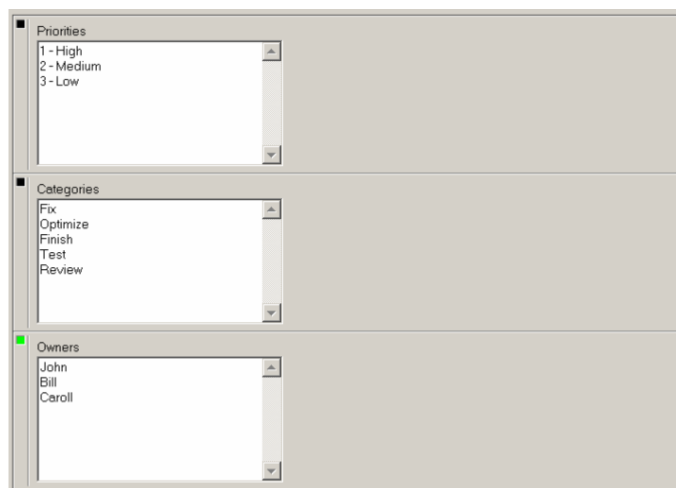
10..20

User Definitions

Item	Value

这个首选项页允许你为数据生成器工具定义默认的全局设置。这些默认设置可以支配每一个数据生成器的定义。要了解更多关于这些首选项含义的详细资料，请看第 15.8 章。

14.23 工具 – 任务列表



这些首选项控制了在任务项目编辑器里的*优先权*、*种类*和*所有者*选择列表里的值。对于优先权，它可以用于给每一个带有数字的值加上前缀，因此这个列表可以用优先权来排序。要了解更多详细资料，请看第 11 章。

14.24 工具 – 重新调用声明



在这个首选项页上，你可以定义声明重新调用缓冲区里声明的最大数和应该被储存的目录。要了解更多详细资料，请看第 16.5 章。

14.25 文件 – 目录

Program files

Test scripts

SQL scripts

Report files

Command files

Macros

Templates

Plug-Ins

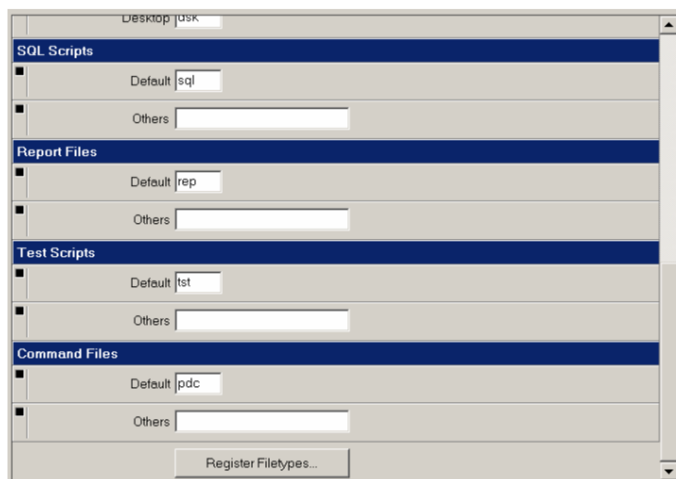
Projects

Standard queries

- **程序文件**
  定义当打开或保存程序文件时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存程序文件的目录。
- **测试脚本**
  定义当打开或保存测试脚本时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存测试脚本的目录。
- **SQL 脚本**
  定义当打开或保存 SQL 脚本时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存 SQL 脚本的目录。
- **报告文件**
  定义当打开或保存报告文件时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存报告文件的目录。
- **命令文件**
  当你从菜单里或通过键入 *@file* 打开一个命令文件时，这个目录将被默认使用。如果你保留这个目录为空，正在工作的目录将被使用。
- **宏**
  这个目录将被用于保存你的宏库。如果你保留这个目录为空，那么 PL/SQL Developer 目录里的 *Macro* **子目录将被使用。**

- **模板**
  这个目录将被用于保存程序文件模板。如果你保留这个目录为空，那么 PL/SQL Developer 目录里的 *Template* 子目录将被使用。
- **插件**
  这个目录将被用于搜索插件。如果你保留这个目录为空，那么 PL/SQL Developer 目录里的 *PlugIn* 子目录将被使用。
- **工程**
  定义当打开或保存工程时的开始目录。如果你保留这个目录为空，你将总是开始于上次打开或保存工程的目录。
- **标准查询**
  定义标准查询被储存的目录。如果你保留这个目录为空，标准查询将被储存于你的概要文件的应用数据目录。

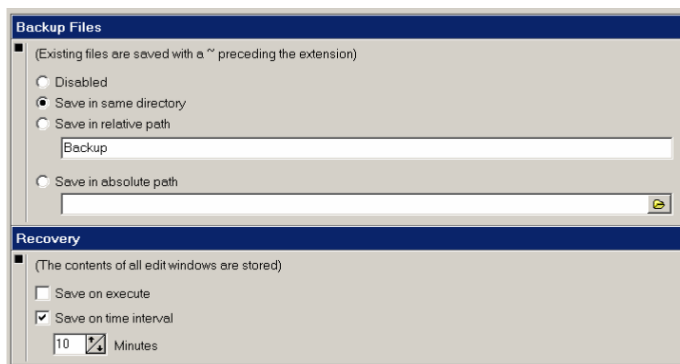
14.26 文件 – 扩展名



在这一页上，你可以定义所有文件类型的默认扩展名。当保存或载入某一文件时，如果你不在文件对话框里指定扩展名，就假定为默认扩展名。对于每一个文件类型，你还可以指定“其它”扩展名，它同样将显示在文件对话框里，多个扩展名可以用逗号来分隔。

通过按*注册文件类型*按钮，你可以把这些默认扩展名与 PL/SQL Developer 联系起来。在 Windows Explorer 里双击带有这样扩展名的文件将自动运行 PL/SQL Developer 。对话框将显示出允许你选择哪一种扩展名与 PL/SQL Developer 联系起来。

14.27 文件 – 备份



这个首选项页包含了针对你的源文件进行备份和恢复的选项。

创建备份文件

在 PL/SQL Developer 保存文件之前，最初的文件在被覆盖以前将被保存到备份文件里。这个备份文件将有同样的名字，扩展名将被加上一个前缀 ~ （例如，employee.pck 将被保存为 employee.~pck）。这样，你将始终有一个早先文件的备份。

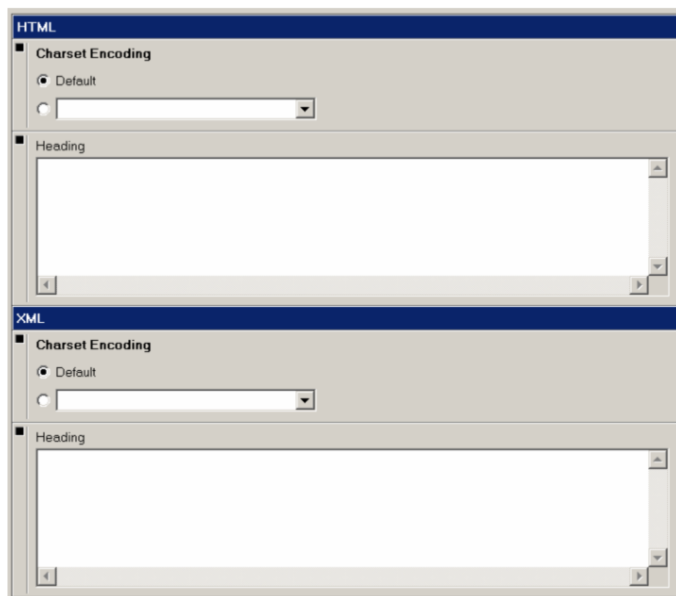
所有的允许你打开文件的文件对话框都将有一个“备份文件(*.~*)”过滤器，它允许你容易地打开备份文件。

恢复

在某一事件上，PL/SQL Developer 可以储存所有打开文件到一个临时位置。万一出现了严重的问题（例如，断电、软件崩溃等），PL/SQL Developer 在随后运行时允许你恢复这些文件。你可以指定下列事件：

- 执行前保存
  在你运行程序窗口、测试窗口、命令窗口、SQL 窗口或报告窗口之前，所有打开的文件都将被储存在一个临时的目录里。
- 定时保存
  所有打开的文件都将在指定的时间间隔被储存在一个临时的目录里。

14.28 文件 – HTML/XML



这些首选项允许你控制由报告窗口和导出功能生成的 HTML 和 XML 输出：

- 字符集编码
  允许你定义 HTML 或 XML 文档的字符集编码。这应该与你的 Oracle 环境的字符集相符，所以数据能被正确地解释。
- 标题
  这个文本将被插入到 HTML 文件的 <HEAD> 部分或者被插入到 XML 文件的标准 XML 标题里。

14.29 其它 – 打印

一般

- **显示打印对话框**
  当按*打印*按钮时，打印对话框将首先被显示出来。
- **打印行号**
  当打印 SQL 或 PL/SQL 源代码时，在每一行之前将显示行号。
- **打印时自动换行**
  比页宽的行将继续在下一行被打印。
- **黑白打印**
  所有被打印的输出都是黑白的，禁止这个选项将创建带有彩色或灰度的输出。
- **打印背景**
  当被允许时，背景颜色将被打印。
- **打印 SQL 窗口里的 SQL 文本**
  决定了当打印 SQL 窗口的内容时在查询结果之前 SQL 文本是否被打印。

表格

- **打印格线**
  决定了表格的格线是否被打印。
- **标题反色**
  决定了表格的标题是否按相反的颜色打印。

页眉

- **标题**
  当被允许时，在页眉的左边，每一页都将有文件类型和文件名被打印。
- **日期**
  当被允许时，在页眉的右边，每一页都将有当前的日期和时间被打印。

页脚

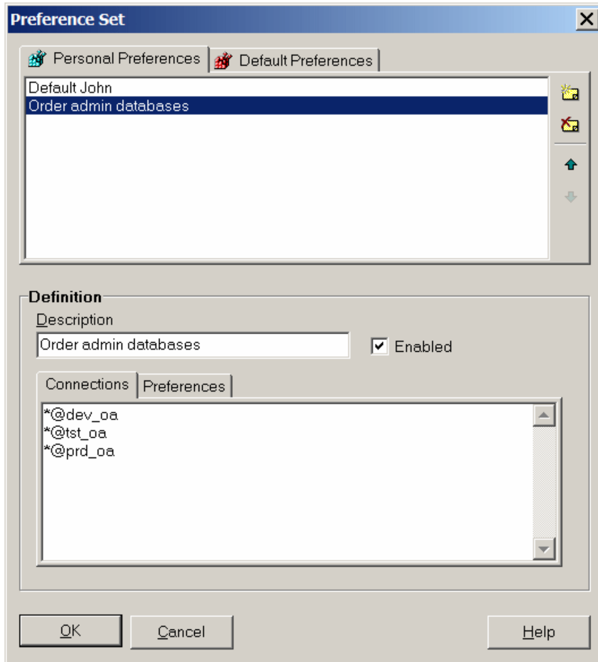
- **页码**
  当被允许时，页脚的中间将包括页码。

14.30 其它 – 更新与消息

允许你控制 PL/SQL Developer 是否和何时检查 Allround Automations 网站的关于 PL/SQL Developer 的更新和消息。如果你禁止这个选项，通过按相应的选项或者通过*帮助*菜单里的*检查在线更新*和*检查在线消息*项目，你可以明白地检查更新和消息。

14.31 首选项集

正如第 14 章所提到的，你可以在多重层级上定义首选项集。要定义首选项，请按首选项集选择列表旁边的*配置首选项集*按钮。下列对话框将显示出来：



*个人首选项*是为当前 Windows 用户定义的（插图里的 *john*）。*默认首选项*可以被系统管理员定义，它为所有的 Windows 用户既在全局层次又在连接层次提供了默认的首选项。

个人首选项

在默认的情况下，所有的用户都将有一个 *Default <用户>* 首选项集，这是你将默认访问的首选项集。要为一个或多个 Oracle 连接创建另外的首选项集，请按首选项列表右边的*新建*按钮。现在，你就可以输入描述和添加一个或多个连接了。对于每一个连接，你可以使用通配符来匹配多用户或数据库。在上面的例子里，*Order admin databases* 首选项集已经被 *john* 定义，用于 *dev_oa* 、*tst_oa* 和 *prd_oa* 数据库。无论 John 何时连接这三个数据库之一，这个首选项集都将被使用。如果首选项没有在这个集里被定义，来自于默认用户的首选项集里的相应的首选项将被使用。

默认首选项

系统管理员可以为所有用户或所有连接定义全局的默认首选项，还可以为一个或多个连接定义默认的首选项集。要做到这一步，请转到*默认首选项*标签页，按首选项列表

旁边的*新建*按钮。如果你不输入任何*连接*，新的首选项集将为所有用户定义默认的首选项。如果你输入了一个或多个连接，首选项集将为这些连接定义默认的首选项。

首选项集优先权

如果首选项是按多层次定义的，则最特殊的层次具有优先权：

1. 个人连接首选项集（最大优先权）
2. 个人 Windows 用户首选项集
3. 默认连接首选项集
4. 默认全局首选项集（最小优先权）

首选项集文件和目录

首选项集被存储在 PL/SQL Developer 安装目录里的 *Preferences* 子目录里的一个单独的文件里。在这个首选项目录里面，个人首选项被存储在以 Windows 用户命名的子目录里。例如：

```
C:\Program Files\PLSQL Developer\Preferences\John
```

默认首选项被存储在 *Default* 子目录里。例如：

```
C:\Program Files\PLSQL Developer\Preferences\Default
```

如果你在一个文件服务器上安装 PL/SQL Developer ，那么你必须提供可读写访问的 *Preferences* 目录、可读访问的 *Preferences|Default* 子目录。作为一种选择，你可以提供 *PREFPATH* 命令行参数来为用户的首选项指定目录：

```
plsqldev.exe prefpath=u:\userdata
```

如果 Windows 用户 *John* 使用 PL/SQL Developer ，那么他的个人首选项集将被储存在 *u:|userdata|john* 里。默认首选项始终被储存在 PL/SQL Developer 安装目录里的 *Preferences|Default* 子目录里。

要保证所有用户都使用一个类似的首选项路径，你可以在 *params.ini* 文件里设置这个参数（请看第 24.4 章）。

15. 工具

在*工具*菜单里有几个工具是可用的，这几个工具将在这一章里被描述。

15.1 浏览器

***浏览器*工具在第 13 章里有详细描述。在*工具*菜单里，通过选择*浏览器*项目，你可以查看或隐藏浏览器。你还可以定义来自于这个菜单的*浏览器过滤器*，它在第 13 章里也有描述。**

15.2 查找数据库对象

这个工具允许你查找基于指定条件的数据库对象，这个条件是指象对象名、类型、所有者或状态这样的条件，另外还有针对对象 DDL 源的全文搜索条件：

你可以在这个表单的上面三部分输入搜索条件。

全文搜索

这个对话框里的选项或多于或少于文本编辑器里的查找对话框的选项，这些选项允许你按不区分大小写搜索、按全字匹配搜索和按正规表达式搜索。如果你忽略了全文搜索，那么符合其它条件的所有对象都将被获得。

对象条件

在这里，你可以输入下列条件：

- **所有者** – 对象的所有者。你可以使用 SQL 通配符（例如 sys%）。默认值为 *<CURRENT USER>*，它将为当前连接的用户限制搜索对象。
- **名称** – 对象名称。你可以使用 SQL 通配符（例如 %common）。
- **状态** – 限制搜索有效或无效对象。
- **创建： 以后/以前** – 限制搜索在指定日期后和前创建的对象。使用日期输入框左边的检验栏来允许或禁止这些条件。

- **修改：　以后/以前 – 限制搜索在指定日期后和前被修改的对象。使用日期输入框左边的检验栏来允许或禁止这些条件。**

对象类型

在这里，你可以选择要搜索的对象类型的范围。

搜索结果

按了*搜索*按钮之后，搜索操作就在后台被执行了，搜索结果列表里将显示出搜索结果。

如果你右击搜索结果，你会看到对象的弹出式菜单。弹出式菜单里的第一个项目是*定位文本*，它在程序窗口里打开源，并将立即使你获得搜索文本的第一个具体值。你还可以通过双击搜索结果里的项目来调用*定位文本*功能。

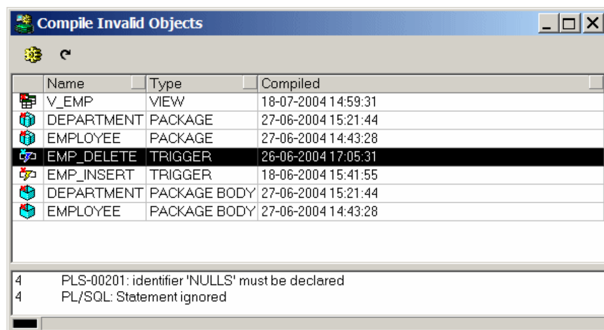
要保存搜索结果，你可以双击它们并从弹出式菜单里的*导出*子菜单项目里选择适当的格式。

保存常用的搜索条件

要保存常用的搜索条件，你可以按*保存*按钮并输入目录和文件名。你以后可以按*载入*按钮来重新调用这些搜索条件。

15.3 编译无效对象

用这个工具你可以容易地重新编译数据库里所有的无效对象。当你选择它时，显示所有无效对象的对话框将显示出来：



注意，当前活动的浏览器过滤器控制了被包括的无效对象。对话框的底部将显示出当前已选对象以前出现的任何编译错误。

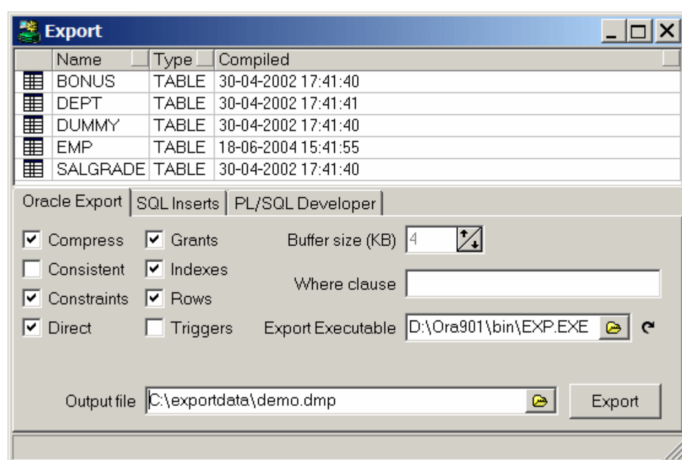
要重新编译对象，请按工具栏里的*编译无效对象*按钮或按主工具栏里的*执行*按钮。对象的依赖关系将被分析以确定编译应该出现的顺序。对象列表左边的图标指示了哪些对象被成功编译了。编译错误部分将在编译完成之后被更新。

要刷新无效对象的列表，请按工具栏里的*刷新对象列表*按钮，这将更新带有当前所有无效对象的列表。

在对象列表里，你可以右击一个对象来访问它的弹出式菜单。

15.4 导出表

导出表工具允许你导出一个或多个表定义和它们的数据到一个文件里，所以你以后还可以导入这些表。在启动了导出表工具之后，你可以选择你要导出的表，选定导出方法（Oracle 导出、SQL 插入或 PL/SQL Developer），设置应用到导出方法的各种选项：



导出功能将分析表之间的可能存在的外键依赖，把表按正确的顺序放到导出文件里。

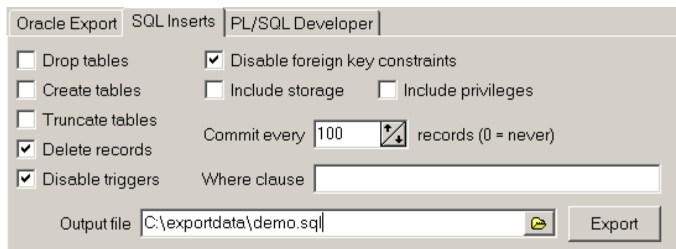
Oracle 导出方法

前一章里的屏幕截图显示了 Oracle 导出方法的选项。这个方法让你指定你可以应用到 Oracle 导出工具的命令行上的选项（请看“ *Oracle 服务器实用工具*”指南），随后将运行它来执行导出。*导出可执行文件*输入框允许你选择一个特定版本的 Oracle 导出工具。在默认的情况下，来自于 Oracle 主目录的最新版本将被使用。

导出完成后，一个新的*日志*标签页将是可见的，这个页包含了被导出工具创建的日志。

这个方法的优势是导出文件的速度和轻便。记住，无论你在何处都可以用 Oracle 导入工具来导入它。

SQL 插入方法

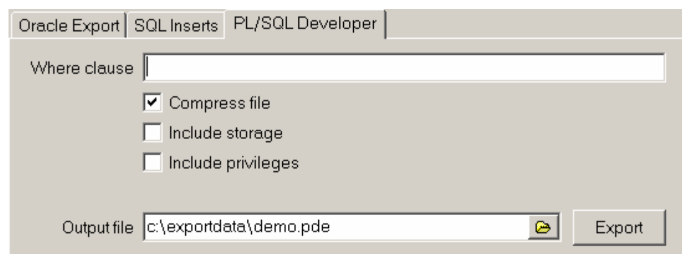


这个导出方法创建了一个标准的带有插入语句和（随意的） DDL 语句的 SQL 脚本来重新创建表定义。标签页包含了下列选项：

- 删掉表 – 已产生的 SQL 脚本包含了一些语句在表被重新创建和载入前来删除掉表。*创建表*、*截短表*和*删除记录*选项在这种情形下被禁止，它们有固有的值。
- 创建表 – 已产生的 SQL 脚本包含了一些语句在表被载入前来创建表。这包括了约束、索引和授权。
- 截短表 – 已产生的 SQL 脚本使用截短表的语句在表被载入前来清空表。这个选项比*删除表*快，但不能被用于外键约束存在的情况下。
- 删除记录 – 已产生的 SQL 脚本使用删除语句在表被载入前来清空表。
- 禁止触发器 – 已产生的 SQL 脚本在表被载入前禁止所有的表触发器，之后允许它们。这可以改善执行性能，有时可能在触发器包含不适用于导入过程的情况下是必要的。
- 禁止外键约束 – 已产生的 SQL 脚本在表被载入前禁止了表的所有外键，之后允许它们。这对于自我参照的外键可能是必要的。两个不同表之间的外键将不导致冲突，这是因为表是按正确的顺序被导出的。禁止外键还可能改善执行性能。
- 包括存储 – 已产生的 SQL 脚本为表创建语句包括了原始存储信息，例如表空间名和初始大小。这些信息可能与数据库不一致，所以这可能并不总是合适的。
- 包括权限 – 已产生的 SQL 脚本包括了对象给其它用户和角色授权的授予权。当对象在不同的数据库里被重新创建时，这些用户和角色显然必须存在。
- 每次提交 – 控制了在提交被执行前有多少条记录将被插入到已产生的 SQL 脚本里。如果这个值是零，在 SQL 脚本的末尾所有的插入将被立刻提交。对于大的导出文件和小的回滚段，在这里给一个适当的值将是必要的。
- Where 子句 – 仅仅匹配 where 子句的记录将被导出。Where 子句必须适用于所有已选的表！

这个方法的优点是 SQL 脚本的轻便和用文本编辑器来编辑结果的可能性。

PL/SQL Developer 方法



这个导出方法创建了定制的、压缩的 PL/SQL Developer 导出文件，它带有表定义和数据。这个标签页包含了下列选项：

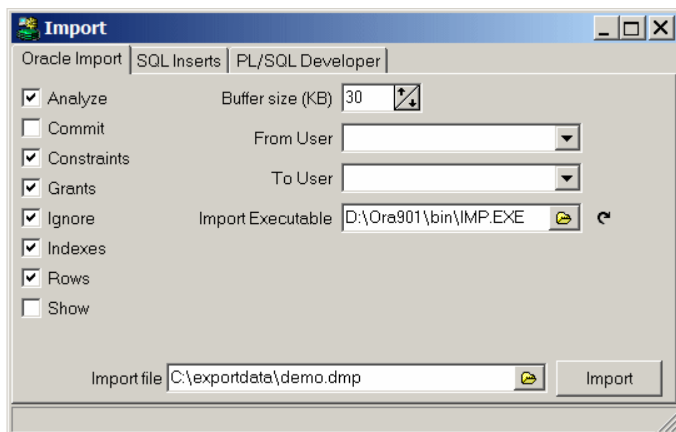
- Where 子句 – 仅仅匹配 where 子句的记录将被导出。Where 子句必须适用于所有已选的表！
- 压缩文件 – 禁止这个选项将禁止压缩。如果数据不能被很好地压缩（例如，long raw 或 BLOB 数据已经在数据库里被压缩），这可能导致执行性能改善。
- 包括存储 – 已产生的 SQL 为表创建语句包括了原始存储信息，例如表空间名和初始大小。这些信息可能与数据库不一致，所以这可能并不总是合适的。
- 包括权限 – 已产生的 SQL 包括了对象给其它用户和角色授权的授予权。当对象在不同的数据库里被重新创建时，这些用户和角色显然必须存在。

这个方法的优点是极大地减小了导出文件的大小，所有的选项都重视导入阶段，在导入期间你可以容易地选择表。

15.5 导入表

导入表工具允许你从用上一章描述的导出表工具在以前导出的文件里导入表定义和数据。正如导出表工具那样，也有三中方法来导入表，每一种都带有它自己的文件格式。

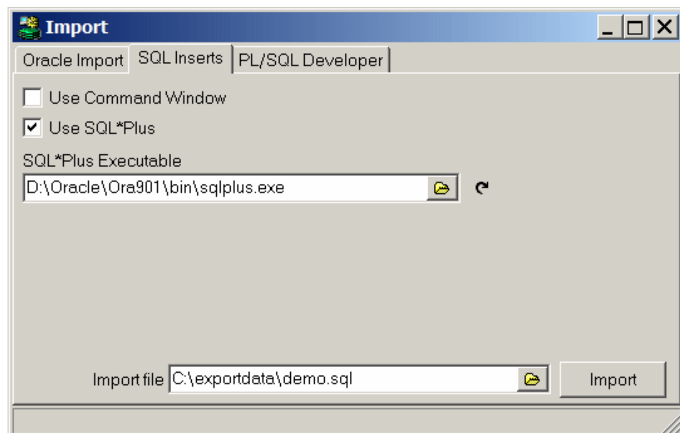
Oracle 导入方法



这个方法使用被 Oracle 导出工具创建的 dmp 文件运行 Oracle 的导入工具来导入数据。你可以制定各种选项，每个选项都要符合导入工具的命令行选项（请看“*Oracle 服务器实用工具*”指南）。*导入可执行文件*输入框允许你选择一个特定版本的 Oracle 导入工具。在默认的情况下，来自于 Oracle 主目录的最新版本将被使用。

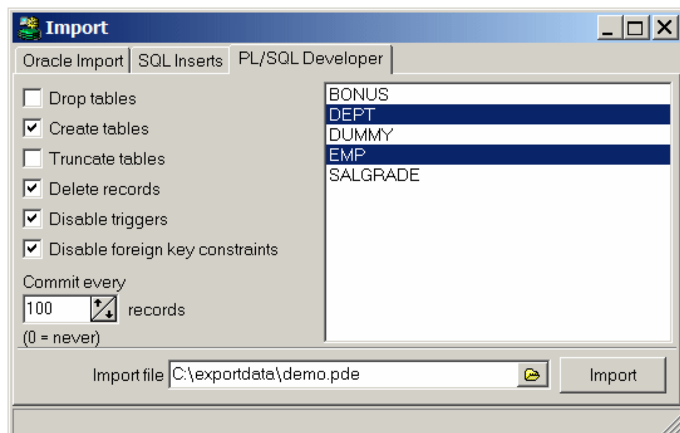
导出完成后，一个新的*日志*标签页将是可见的，这个页包含了被导入工具创建的日志。

SQL 插入方法



这个方法是执行用导出表工具生成的 SQL 脚本。你可以选择启动 SQL*Plus 来运行这个脚本或者使用内置的命令窗口。对于 SQL*Plus ，作为一种选择，你可以选择一个特定的 SQL*Plus 可执行文件。在默认的情况下，来自于当前 Oracle 主目录的版本将被使用。

PL/SQL Developer 方法



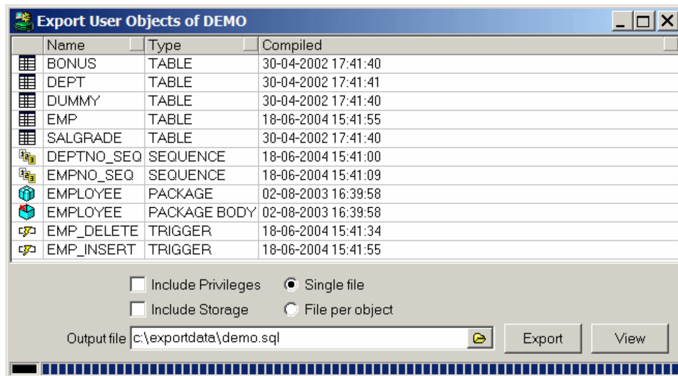
在选择了以前使用 PL/SQL Developer 方法导出的导入文件之后，你可以从列表里选择导入一个或多个文件。下列选项可以被指定：

- 删掉表 – 在表被重新创建和载入前删除掉表。*创建表*、*截短表*和*删除记录*选项在这种情形下被禁止，它们有固有的值。
- 创建表 – 在表被载入前创建表。这包括了约束、索引和授权。

- 截短表 – 在表被载入前截短表。这个选项比*删除表*快，但不能被用于外键约束存在的情况下。
- 删除记录 – 在表被载入前删除表里的所有记录。
- 禁止触发器 – 在表被载入前禁止所有的表触发器，之后允许它们。这可以改善执行性能，有时可能在触发器包含不适用于导入过程的情况下是必要的。
- 禁止外键约束 – 在表被载入前禁止表的所有外键，之后允许它们。这对于自我参照的外键可能是必要的。两个不同表之间的外键将不导致冲突，这是因为表是按正确的顺序被导出的。禁止外键还可能改善执行性能。
- 每次提交 – 控制了在提交被执行前有多少条记录将被插入到已产生的 SQL 脚本里。如果这个值是零，在 SQL 脚本的末尾所有的插入都将被立刻提交。对于大的导出文件和小的回滚段，在这里给一个适当的值将是必要的。

15.6 导出用户对象

要导出一个用户的所有对象的 DDL （数据定义语言）语句，你可以使用*导出用户对象*工具。这样你可以容易地为另一个用户或者在不同的数据库里重新创建对象。选择了来自于*工具*菜单的*导出用户对象*之后，当前用户的所有对象都将呈现在这个表格里：



在选择了一个输出文件之后，你可以按*导出*按钮来导出对象。如果没有在表格里选择特定的对象，所有的对象都将被导出。你可以通过在它们上面点击来选择对象，使用 *Ctrl* 和 *Shift* 键来选择多个对象和对象范围。

输出文件是一个与 Oracle 的 SQL*Plus 和 PL/SQL Developer 的命令窗口兼容的 SQL 脚本。在导出操作之后，你可以按*查看*按钮在命令窗口里打开已产生的文件。

如果你要包括对象给其它用户和角色授权的权限，*包括权限*选项可以被允许。当对象在不同的数据库里被重新创建时，这些用户和角色显然必须存在。

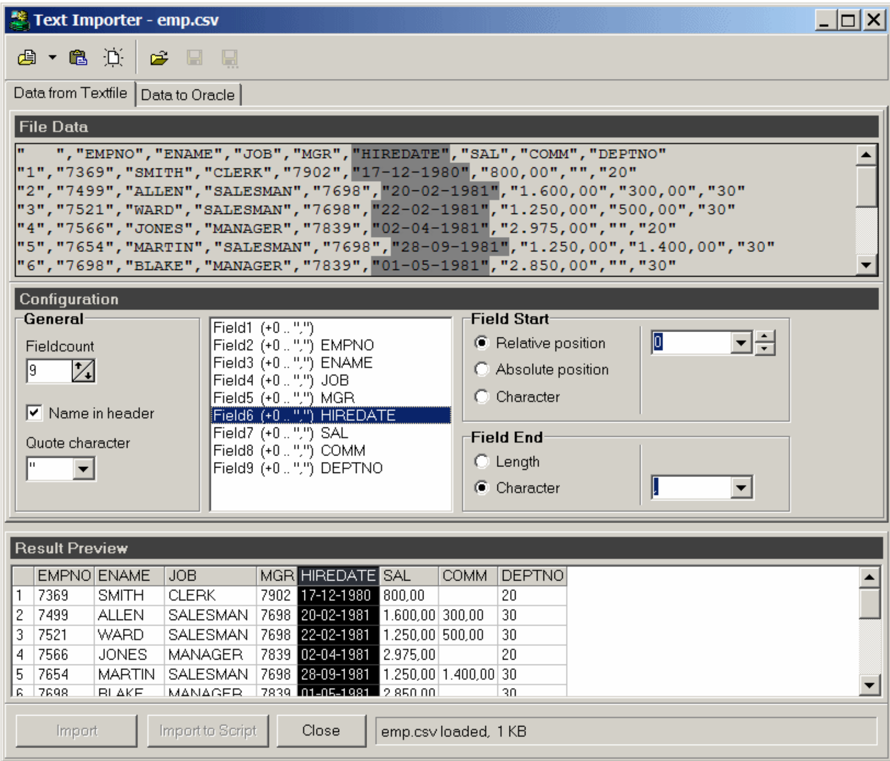
同样地，你可以允许*包括存储*选项来包括存储信息，例如表空间名和初始大小。这些可能与数据库不一致，所以这可能并不总是适当的。

单个文件和***每个对象的文件***选项控制了包含了所有对象的 DDL 的单个 SQL 脚本是否被创建，或者每个对象是否应该被导出到一个特定的文件（对象名为文件名、.sql 为扩展名）。那样的话，输出文件将包含一连串的对这些对象特定文件的调用。所有文件都将被写到输出文件的目录里。

15.7 文本导入器

使用文本导入器你可以把 ASCII 文件导入到数据库里。象用逗点和制表符分隔的那样的被标定格式的大多数行都是被支持的。导入器将设法自动确定文件的格式，所以大多数时间里你不需要定义任何事情，仅仅选择文件，选择表，如此而已。

当你打开文本导入器并载入文本文件时，你将获得如下窗口：



导入器从文件里载入了前 100 行的数据，原始数据显示在窗口上部的“文件数据”里。中间部分允许你制作文本文件的定义。下部的“结果预览”有一个带有将被导入的数据的表格。

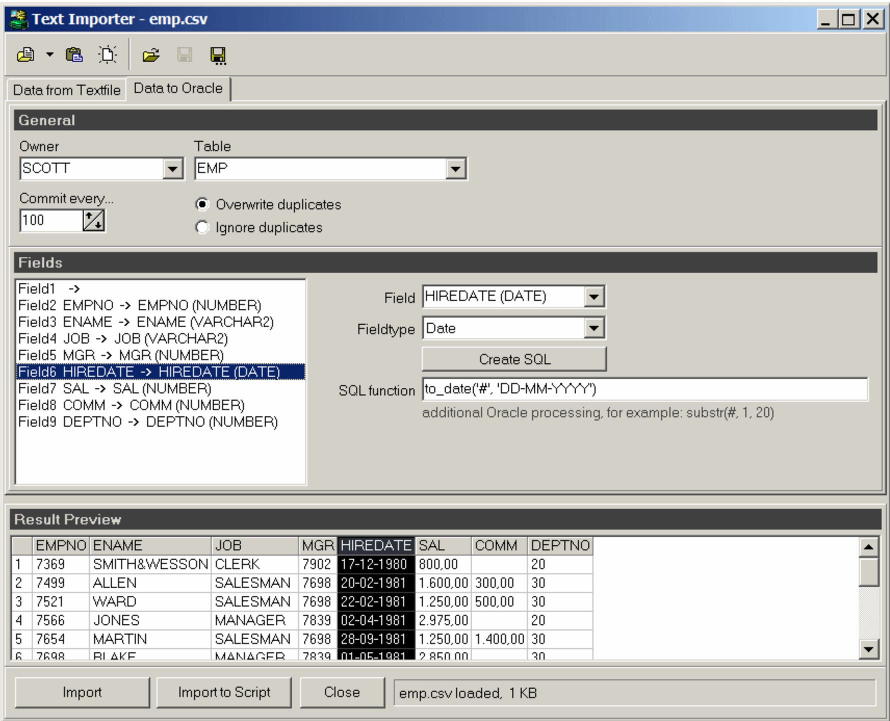
工具栏里有按钮用于选择文本文件或从剪贴板里粘贴文本。文件可以实际上不限制大小，它们不是被完整地读到内存里。“新建”按钮将清空文本导入器，打开和保存按钮将允许你重新使用定义。

你一载入文件，配制就被自动确定，如果这不正确，你自己可以创建和修改它。把字段计数设置为正确的数字，从字段列表里选择字段来定义字段的定义。在文件数据部分，被指定的字段被高亮显示，这允许你检查字段定义是否是正确的。你有下列的选项来配置文本导入器的定义：

- **一般 – 字段计数**
  每行文本的字段数量。
- **一般 – 标题名**
  指出第一行文本是不是字段名。
- **一般 – 引号字符**
  指出确定字符串的字符，通常为双引号或单引号。
- **字段开始 – 相对位置**
  本字段的首端与前一个字段的末端的相对位置。0 表示前一个字段的末端就是本字段的开始位置。
- **字段开始 – 绝对位置**
  这指出字段开始于一个固定位置。
- **字段开始 – 字符**
  这指出字段开始于一个特定的字符，这与前一个字段的末端有关。
- **字段结束 – 长度**
  指出字段有一个固定的长度。
- **字段结束 – 字符**
  指出字段结束于一个特定的字符。

对于标准的逗号分隔的文件（csv），所有的字段的开始相对位置都是 0 ，结束这个字段的结束字符应该是逗号。行结束符（cr/lf 或 lf 或 cr）也将指出字段的结束，所以对于一行上的最后一个字段，结束字符并不是重要的。

当你有一个正确的文件定义时，你需要选择一个 Oracle 表并指出哪个文本字段应该插入到哪个 Oracle 字段里。你可以在第二个标签页上做这件事：



	EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
1	7369	SMITH&WESSON	CLERK	7902	17-12-1980	800,00		20
2	7499	ALLEN	SALESMAN	7698	20-02-1981	1.600,00	300,00	30
3	7521	WARD	SALESMAN	7698	22-02-1981	1.250,00	500,00	30
4	7566	JONES	MANAGER	7839	02-04-1981	2.975,00		20
5	7654	MARTIN	SALESMAN	7698	28-09-1981	1.250,00	1.400,00	30
6	7698	BLAKE	MANAGER	7839	01-05-1981	2.850,00		30

上部有一些一般的导入参数，“字段”部分允许你在文本文件字段与 Oracle 字段之间建立联系，下部是你的信息的结果预览。

你可以选择下列的一般首选项：

- 表
  你要导入进数据的 Oracle 表（或视图）。在选择了表之后，导入器将设法自动地确定字段和字段类型。这仅仅工作于文本文件有字段标题的情况下。
- 每次提交…
  指出每次你要提交多少条纪录。如果你设置它为 0 ，在整个导入的末尾所有的数据都将被提交。
- 覆盖重复
  重复的记录将在数据库里被更新。
- 忽略重复
  重复的记录将被忽略。

在字段定义里，你能看到一个来自于文本文件带有字段的列表。对于每个字段，你可以设置如下：

- 字段
  你要与文本字段建立联系的 Oracle 字段。如果你不要求导入这个字段，就可以保留这个字段为空。
- 字段类型
  基本的字段类型：字符串、数字或日期。
- SQL 函数
  这个选项允许你定义附加的 SQL 处理。对于日期字段，to_date 函数被自动添加。这是一个很强大的选项，允许你转换导入的数据。基本上，你可以输入能被 Oracle 处理的任何东西。你可以添加一个 # 号来指出数据。“创建 SQL”按钮将为日期字段用 to_date 函数来填充这个输入框。

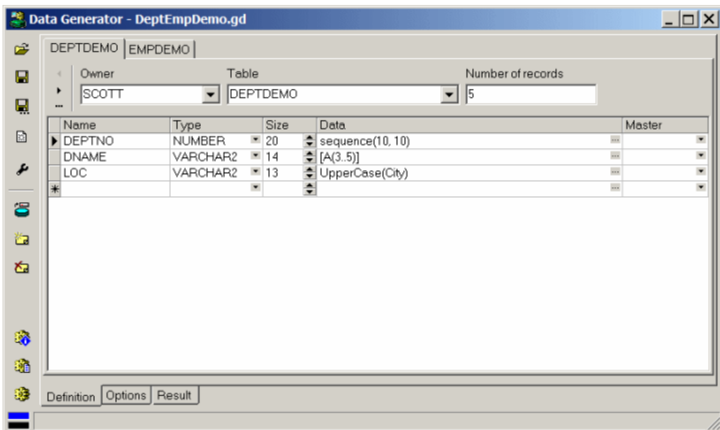
当定义完成时，你可以决定保存它，工具栏里有保存和载入定义文件的按钮。文本导入器将记住哪一个定义文件被用于哪一个文本文件，并且在下一次你打开同样的文本文件时，定义文件将自动被载入。

你有两个导入按钮：“导入”和“导入到脚本”。第一个选项将开始导入数据到已选的表里，第二个选项将创建一个带有插入语句的 SQL 脚本。

15.8 数据生成器

数据生成器允许你创建演示和测试数据。对于测试应用程序和查看它们怎样执行大量的数据，这可能是有帮助的。你可以在工具菜单的下面找到数据生成器。

基本上，定义由一个或多个表、你要生成的记录数量和字段数据定义构成。左上部的按钮允许你打开和保存定义。



在上面，你可以看到被提供的 deptemp 演示，这个演示是为 deptdemo 和 empdemo 表创建数据，这类似于大家都知道的 dept 和 emp 表。在这一页里，你将发现下列项目：

- 表 – 表的名称。

- 记录数 – 你要生成的记录数。这可以是数字，也可以是象 10..100 这样的范围。
- 名称 – 字段名称。
- 类型 – 字段的数据类型。
- 大小 – 字段大小（在适当的时候）。对于数字字段，这将是数值范围、精度。
- 数据 – 字段数据的定义（见下面）。
- 主列 – 如果这个表是另一个表的明细，你可以设置主列。对于每一个已产生的记录，指定数量的明细记录将被生成。

你可以使用“添加现有表”按钮来添加一个现有的表，你还可以从对象浏览器里拖放一个表。

数据定义

数据定义决定了被生成的数据。如果你要创建简单的字符，你可以在两个方括号之间输入字符定义：[数据]

数据可以是下列预先确定的集的混合体：

- a: a..z *(小写字符)*
- A: A..Z *(大写字符)*
- @: A..Z 和 a..z *(所有字符)*
- #: A..Z 和 a..z 和 0..9 *(所有字符和数字)*
- *: #33..#*126（所有 ASCII 字符）*
- 0: 0..9 *(所有数字)*
- 1: 1..*9（所有除了 0 之外的数字）*
- 9: 0..9 *(所有数字)*

例如：

[Aaa00] 生成字符串如： Gxe21，Liy05 等等。

你还可以在两个单引号之间添加文字文本。

例如：

[AA '-' 1000] 生成字符串如：CX-4903 ，SY-1044 等等。

在定义里空格字符是被忽略的，除非它们在引号里。

如果你要一个字符重复许多次，你可以在字符的后面两个括号之间添加次数(n)。你还可以添加一个随机数(最小..最大)。

例如：

[Aa(5..15) ' ' Aa(8..20)] 的结果就象这样："Masfae Qwwecdsadif"

定义里的文字文本不必被装入到括号里。换句话说，['hello'] 和 'hello' 是等效的。不带引号被输入的文本被认为是函数。

有几个特定的函数是可用的：

- Signal(Min, Max, Delta, Noise)
  返回技术测定数据（象温度这样的）。Min（最小）和 Max（最大）决定了范围， Delta（△）为最大改变量，你还可以添加一些 Noise（噪声）。例如：Signal(-10, 20, 0.1, 0.1)。
- Random([Min], Max)
  返回 Min（最小）和 Max（最大）之间的随机数。如果只指定了 Max ，那么 Min 就被设为 0 。对于日期字段，你可以输入 Min（最小）和 Max（最大）的日期。
- Sequence(Start, [Inc], [WithinParent])
  返回序号。Start 为开始值，Inc 为增量（默认为 1）。对于明细表，你可以再指定 WithinParent 关键词来指出序列应该为每一个父记录进行重新安排。
- List('item'(weight), 'item'(weight), ...)
  随机地返回指定项目（item）之一。加权数（weight）可以被添加到两个括号之间，允许特定项目比其它项目出现的几率更大。
  例如：List('CLERK'(50), 'SALESMAN'(30), 'MANAGER'(10))
- List(select statement)
  象前面的 List 函数，但项目是通过 SQL select statement（select 语句）返回的。
- Text([Style], MaxCharacters, [WordsPerLine, LinesPerParagraph])
  这个函数返回文本。可选择的 Style（字体）参数可以是 LorumIpsum（默认）（怀疑是荷兰语。——译者姜华东注）、英语、德语或日语。它还可以是象 [aA] 这样的字符集，在这种情况下，词汇从指定的字符集里产生。
  MaxCharacters 决定了生成的文本的最大大小，WordsPerLine 和 LinesPerParagraph 决定了行和段落的大小。指定的大小可以是指定的数字或范围（最小..最大）。
- File(path, path, ...)
  从特定的路径（path）里随机地选择一个文件，并插入内容。允许你输入二进制数据（象图片这样的）到数据库里。路径可以使用通配符，象 d:\images*.bmp 这样的。

你可以通过使用下列函数改变前面函数的文本结果：

- Uppercase()
- LowerCase()
- InitCaps()

例如：InitCaps(List(select ename from emp))

还有几个预先定义的数据集可用于生成或多或少的真实数据。你可以使用下列定义：

- Firstname – 一般列表里的名
- Lastname – 一般列表里的姓
- Company – 公司名（现有公司的随机列表）
- Address1 – 地址行 1
- Address2 – 地址行 2

- ZIP – 邮政编码
- City – 城市
- State – 省
- Country – 国家
- Email （被关联到 Firstname ，Lastname 和 Country）

还有一些可用的实例自定义数据集：

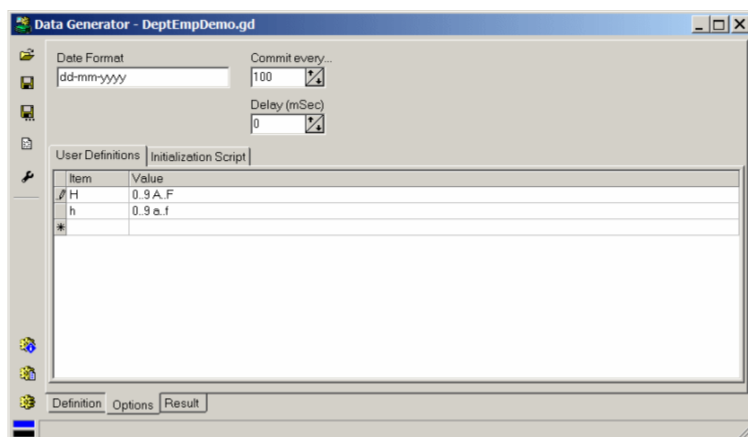
- Components.Code – 一般商品项目：商品代码
- Components.Description – 商品描述（计算机部分）
- Components.Price – 商品价格
- Elements.Name – 化学元素（名称）
- Elements.Symbol – 化学元素（符号）

这些数据集可以在 DataGenerator\UserData 目录里被找到，文件名为 elements.txt 和 components.txt 。如果需要，你可以添加自己的集。添加一个逗号分隔的文件很简单，在文件里第一行两个方括号之间保存着描述。你可以通过指定“文件名.描述”来使用你的文件里的数据，就象两个例子那样。

上面提及的所有函数和数据都可以添加到一起，例如：Random(10..99) + '-' + [A(4)] 。
“+”是可选的，但那里至少应该有一个空格作为分隔符。

选项

选项标签页允许你设置一些首选项，特别是数据生成器的定义。你可以指定一个日期的格式（与被 Random 函数使用的例子一样）。你还可以指定在多少条记录之后需要提交（设为 0 是一完成就提交）。延迟首选项仅仅用于一些实时测试，在这里你需要数据以指定的速度被插入。



自定义部分允许你定义或越过被用于［数据］定义的字符集。项目总是单个字符，值是一个或多个空格分隔的字符范围，就象这样：A..Z a..z 0..9 #200..#220 。例如，要定义一个十六进制的字符集，你可以指定 H 为项目，0..9 A..F 为值。

“初始化脚本”部分允许你指定在数据被插入到数据库里前应该被执行的 SQL 语句。在典型的情况下，这将是创建表或截短表、选择回滚段等等的语句。多个语句需要用分号来分隔。

通过按左边*首选项*按钮，在默认的情况下，对于所有数据生成器的一般选项可以被定义。要了解更多详细资料，请看第 14.22 章。如果数据生成器定义里相应的首选项也被设置了，这些首选项将被否决。

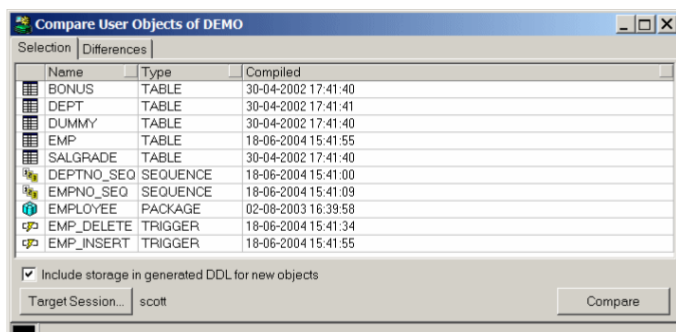
生成数据

在左下方有三个按钮可用于生成实际的数据：

- 开始测试运行 – 这将在*结果*标签页上的表格里生成数据和显示结果。通过右击表格，你可以以不同的格式导出这些结果。
- 创建数据为 SQL – 生成数据为 SQL 脚本。这仅仅工作于你不用文件功能添加来自文件的数据的情况下。
- 在数据库里创建数据 – 生成数据到 Oracle 数据库里。

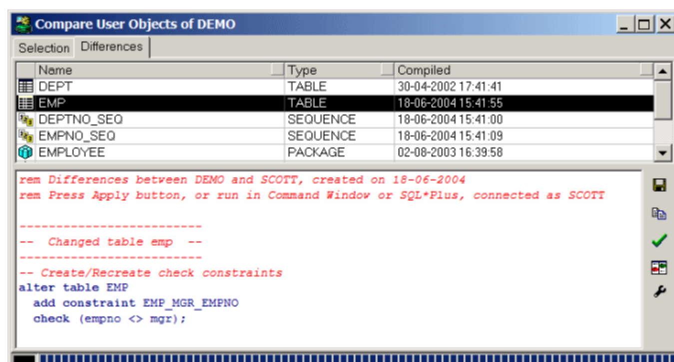
15.9 比较用户对象

在表定义、视图、程序单元等发生改变之后，把这些改变传递到另外的数据库用户也许是有用的。这也许是另外一个开发环境或者是一个测试环境。要把你的开发用户对象与另一个用户对象进行比较，你可以使用*工具*菜单里的*比较用户对象*功能。这将调出下列对话框：



在*选择*标签页里，你可以选择要比较的对象。在做好了选择以后，你可以按*目标会话*按钮，选择你要比较的用户和数据库。这将*激活*比较按钮，你可以按它来开始比较操作。你可以选择*包括存储…*选项来包括存储信息，如新对象的表空间名和初始大小。这些可能与数据库有不同的地方，所以这并不总是适用的。

当比较操作完成后，对话框将转到*差异*标签页，它将显示出所有有差异的对象的列表：



这个列表按依赖的顺序被储存。在目标用户不同对象的列表下面，你可以看到需要被执行的 SQL 语句，这些语句能使这些对象与当前用户的相应对象变得一致。如果没有对象被选择，所有对象的 SQL 都被显示。如果一个或多个对象被选择，那么已选对象的 SQL 就被显示。在上面的例子里，丢失的 *EMP_MGR_EMPNO* 检查约束就被添加到 *EMP* 表里了。

*显示差异*按钮将显示出对象的新旧源文件之间的每一个差异。这对于你查看程序单元里产生的改变可能是有用的，或者它可以帮助你来确定对于其它对象类型一个特定的 DDL 语句为什么会被生成。*配置外部比较工具*允许你配置应该被使用的比较工具。在默认的情况下，*ExamDiff* 工具将被使用，Pro 版本是可用的（请看 ExamDiff 的 *info* 菜单的*关于*项目）。请参看第 14.21 章。

现在，你可以按*应用 SQL* 按钮在目标会话里执行这个 SQL 了。作为一种选择，你可以通过按*保存 SQL* 按钮在一个文件里保存 SQL ，或者你可以通过按*复制 SQL* 按钮把它复制到剪贴板上。

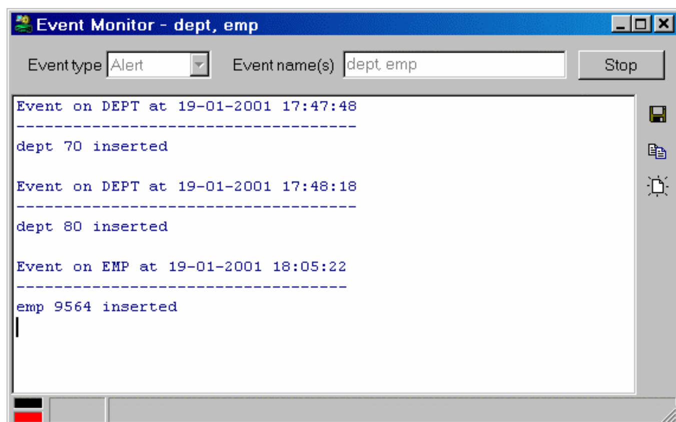
当对象被比较时，下列属性会被忽略：

- 储存 – 对于比较，象表和索引的*下一个大小*和*%空闲*这样的属性不被做相应的考虑。
- 带有系统已生成名称的约束 – 对于两个用户，这些约束将有不同的名称，所以它们不能被比较。如果表在目标会话里是新的，这些约束将被生成。
- 表创建属性 – 需要重新创建表的属性被忽略。
- 表数据 – 要导出表数据，请使用*导出表*功能（请看第 15.4 章）。
- 序列值 – 序列的当前值是被考虑的数据。

15.10 事件监视器

事件监视器允许你捕获和查看被在其它会话里的程序单元生成的 dbms_pipe 信息或 dbms_alert 信号。仅仅指定*事件类型*（Alert（报警）或 Pipe（鸣叫））、报警和鸣叫的*名称*，再按*开始*按钮就可以。

下列屏幕截图显示了事件监视器在等待名为 ‘DEPT’ 或 ‘EMP’ 的警报：



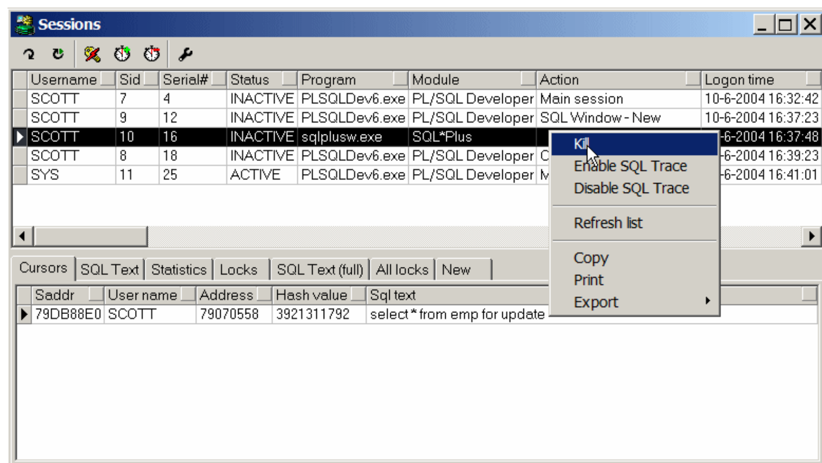
无论这样的事件何时出现，鸣叫或警报的名称和时间戳都被写到输出页里。事件的信息如下：

- 鸣叫：信息的所有项目都被显示出来。
- 报警：dbms_alert.signal 呼叫的信息参数的值被显示出来。

对于报警事件，你可以指定多个名称，用冒号来分隔。对于鸣叫事件，你可以只指定一个鸣叫名称。

15.11 会话

会话工具显示了你当前连接到的数据库实例里所有的会话：



在窗口的上半部分里，你可以看到关于会话的信息，例如用户名、sid 和序列号（用于识别会话）、状态等等。你可以通过点击列标题按钮给会话列表排序。例如，要快速查找所有活动的会话，请点击状态列标题按钮。

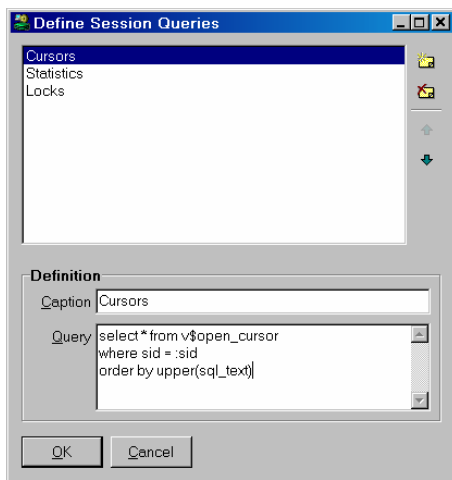
如果你右击一个会话，这将调出一个带有下列项目的弹出式菜单：

- 关掉 – 允许你关掉已选的会话。
- 允许 SQL 跟踪 – 允许 SQL 跟踪已选的会话。如果你的 Oracle 服务器版本不支持这个特性，或者如果你无权使用 dbms_system 包，这个选项将被禁止。
- 禁止 SQL 跟踪 – 禁止 SQL 跟踪已选的会话。
- 刷新列表 – 刷新会话列表。
- 复制 – 复制会话列表到剪贴板。
- 打印 – 打印会话列表。

刷新、关掉、允许 SQL 跟踪和禁止 SQL 跟踪功能通过窗口顶部的工具栏也是可访问的。工具栏还包括了*自动刷新*按钮。当它被按时，会话列表和会话信息将定时被更新。你可以通过右击按钮来定义刷新周期。

窗口的下半部分包含了关于已选会话信息的几个标签页。通过右击列表并选择来自于弹出式菜单的相应项目，这些标签页上的信息将可以被刷新、复制或打印。要了解更多关于显示在列表里的单独列的信息，请看“*Oracle 服务器参考*”指南。

对于会话信息标签页，你可以定义你自己的查询，或者修改标准的查询。要做到这一点，请按工具栏里的*定义会话查询*按钮。这将调出下列对话框：



在右边，你可以看到四个按钮：创建新查询、删除查询或在列表里上移、下移查询。当你创建或修改查询时，你必须提供下列信息：

*标题*揭示了会话窗口的标签页。列表里的位置决定了会话窗口里的标签位置。

*查询*是提取会话信息的 SQL select 语句。这个 select 语句通常应该包含 *:sid* 绑定变量，当查询被执行时它将包含已选会话的 sid 。你可以使用 v$session 视图的列中的任何一个列，例如 *:username* ， *:sql_address* 或 *:sql_hash_value* 。如果你的查询导致运行时间错误，它将在执行后相应的标签页上显示出来。

在上面的例子里，*Cursors* 查询显示了已选会话的所有打开的游标，按 SQL 文本排序。

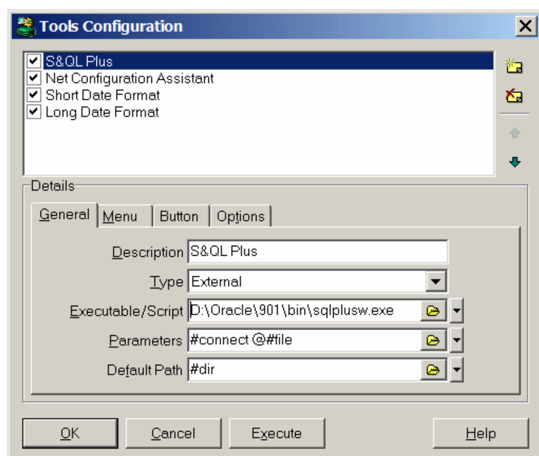
15.12 自定义工具

你可以定义你自己的集成在 PL/SQL Developer IDE 里的工具。这些工具可以被添加为菜单项目，用于为当前会话在数据库里启动外部的应用程序或执行 SQL 或 PL/SQL 脚本。你可以配置参数，所以对于应用程序你可以做到忽略文件名和连接字符串。

这样做的一个很好的例子是 SQL*Plus 。你可以添加一个菜单来启动 SQL*Plus 并且带有适当的配置，这样你能让 SQL*Plus 自动登录，甚至让它执行当前打开的文件。

另外一个例子是为 NLS 设置设一套特定的值的工具。你可以创建一个脚本来执行正确的“警报会话”语句，并且添加这个脚本到菜单里，在菜单里它可以被调用用于 SQL 窗口、测试窗口等的会话。

要配置外部工具，你就必须选择*工具*菜单里的*配置工具*菜单项目。下列对话框将显示出来：



右边的四个按钮允许你插入和删除项目，还可以在列表里上移或下移项目。如果你在按插入按钮时按了 Ctrl 键，新的已创建的项目将从当前的选择里被复制。

执行按钮可以被用于执行已选的工具。如果这个按钮被选择时你按下了 Ctrl 键，一条信息将弹出来，它包含了关于应该执行什么的（带有替换参数的）信息。

列表里显示出了所有的已配置的工具，下半部分显示了已选工具的配置。配置被分为四部分：

- 一般 – 定义被运行的可执行文件/脚本和它的参数。
- 菜单 – 定义相应的菜单项目应该显示的位置。
- 按钮 – 为工具栏定义按钮图片和描述。
- 选项 – 一些附加的工具设置。

要解释外部工具配置，我们将把添加一个 SQL*Plus 菜单到 PL/SQL Developer 作为一个例子来说明问题。

一般标签页

在你用插入按钮创建了一个新项目之后，第一件要做的事情是定义工具类型。如果你选择了*外部*，工具将被作为一个外部应用程序来运行。这对于 SQL*Plus 来说是可用的。如果你选择了*会话*，当工具被调用时，工具将运行当前会话的 SQL 脚本。

其次，你要在一般标签页上输入描述。描述是将被显示在菜单和列表里的名称。你可以在要作为快捷键的字符前面输入一个 & 符号（S&QL*Plus 会变成 SQL*Plus）。如果你输入了 – 符号作为描述，一个分隔行将被显示出来。

第三个至关重要的事情是用于执行的程序或脚本的名称。对于外部工具，你可以输入任何可执行文件甚至是你希望的文档，对于文档与其关联的应用程序将被启动。浏览按钮打开了一个文件对话框，允许你查找文件。大多数 Oracle 工具都位于 Oracle 的 bin 目录里，SQL*Plus 可以被发现于 <oracle_home>\bin\sqlplusw.exe 。

要使 SQL*Plus 能从菜单里工作，这三个步骤已经足够了。另外，你可以定义一些参数传递给应用程序，还可以定义默认路径。你可以在默认路径和完整路径之间选择一个来用于可执行文件或文件参数。对于 SQL*Plus ，你可以添加一个连接字符串（#connect）和一个要执行的文件的连接（#file）。

右边带有下箭头的小按钮允许你挑选一个变量，这个变量能被插入到输入框的任何地方。这些变量都与当前连接和已打开的文件有关系。当工具运行时，它们将被相应的信息替换。下列变量是被支持的：

变量	含义
#file	当前窗口里不带路径的文件名
#path	带路径的文件名
#dir	当前窗口里文件的目录
#object	已选的浏览器对象（如 SCOTT.EMP ）
#otype	已选的浏览器对象类型（如 TABLE ）
#oowner	对象所有者（SCOTT）
#oname	对象名（EMP）
#connect	当前 PL/SQL Developer 连接的完整连接字符串（如 scott/tiger@demo）
#username	用户名
#password	口令
#database	数据库

SQL*Plus 不喜欢可能以空格为文件参数的完整路径，它被关掉了，这就是为什么 #dir 被指定为默认路径的原因，这样仅仅文件名必须被传递到 SQL*Plus 。

会话工具

对于会话工具，你要指定一个能包含多个 SQL 语句或 PL/SQL 块的 SQL 脚本。SQL 语句能被分号或斜杠分隔开，PL/SQL 块必须用斜杠来终止，这与 SQL*Plus 的语法一样。

例 1 – Germany.sql:

```
alter session set nls_language = 'german';
alter session set nls_territory = 'germany';
```

例 2 – SpecialRole.sql:

```
begin
  dbms_session.set_role(role_cmd => 'special_role identified by &Password');
end;
/
```

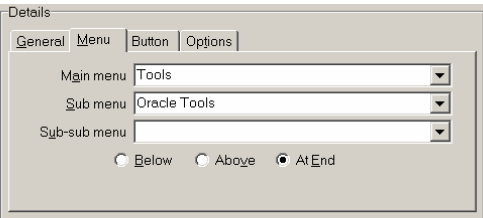
注意，你可以添加置换变量到脚本里以使它变得更为灵活。当工具被调用时，你会得

到提示要输入变量的值。当上面的 SpecialRole 脚本被调用时，你将被提示要求输入一个口令。

如果你要使用任何一个脚本里指定的参数，那么你就必须使用 &1、&2 等等。数字指出了命令行上参数的顺序。你将不会得到这些命令行置换变量的提示。

菜单标签页

在默认的情况下，所有被配置的工具都被加到工具菜单的底部。如果你要在别处甚至它们自己的主菜单里创建它们，你就可以使用菜单标签页：



如果你要求你的工具菜单项目被创建到特定的主菜单里，你可以从主菜单编辑列表里选择一个项目。或者如果你要创建一个新的主菜单，那么你就可以输入一个新的菜单名。

如果你已经输入了一个主菜单，那么你就可以使用子菜单和子子菜单来指出你要创建的新菜单项目的准确位置。

你可以使用三个无线按钮（在下面、在上面和在末端）来指定新菜单应该在指定的相关菜单的什么地方。如果你指定了一个新的（子）菜单，你应该始终选择"在末端"，因为"在下面"和"在上面"仅仅指的是你涉及到的现有（子）菜单的下面和上面。

例 1:

主菜单	工具
子菜单	Oracle 工具
子子菜单	
位置	在末端

这将在工具主菜单里创建一个 Oracle 工具子菜单。如果你正在添加多个 Oracle 相关工具，那么设一个特定的子菜单可能是一个好注意。

例 2:

主菜单	Oracle
子菜单	
子子菜单	
位置	在末端

这将创建一个 Oracle 主菜单。如果你正在有规律地使用一些工具并且你要求它们"在附近"，那么你大概应该只好创建一个新的主菜单。

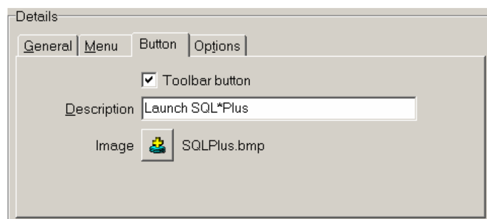
如果你要添加你的工具为工具菜单里的第一个项目，那么你应该输入：

主菜单	工具
子菜单	首选项…
子子菜单	
位置	在上面

这将在现有的首选项菜单项目上面创建一个菜单。

按钮标签页

你可以定义一个外部工具是否应该和应该怎样被包括到工具栏里：



- **工具栏按钮**

  当这个选项被允许时，外部工具就可以被包括到工具栏里了。

- **描述**

  当你控制着鼠标光标越过工具栏按钮上方时，描述将作为提示被显示出来。如果你保留这个描述为空，外部工具的描述（与显示在菜单里的一样）将被使用。

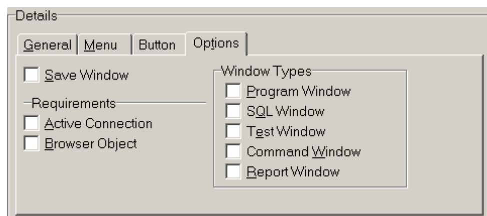
- **图象**

  按图象按钮来为工具栏按钮选择一个 Windows 位图文件（*.bmp）。这个图象的大小应该是 20 x 20 像素比较合适。注意，PL/SQL Developer 将始终从原始位置载入位图文件，因此若相应的外部工具不改变你就不要移除或重新命名这个文件。

  PL/SQL Developer 带有许多标准的位图文件，你可以从那里挑选它们。这些文件位于 PL/SQL Developer 安装目录里的 Icons 子目录里。这是图片选择器默认的目录。

选项标签页

选项标签页允许你进行下列设置：



- **保存窗口**

  当这个选项被设置时，活动窗口将在工具被执行前被保存。当工具获得作为参数的文件并且你要确信它使用当前数据时，你大概应该做保存窗口这件事。

- **激活连接**

  如果工具正好用于要求 PL/SQL Developer 被连接（如果你传递作为参数的连接字符串），你就应该设置激活连接了。这将在 PL/SQL Developer 没有被连接时禁止使用菜单。

- **浏览器对象**

  如果工具工作在浏览器里已选的对象上，就设置这个选项。

- **窗口类型**

  如果工具需要一个特定的窗口，你可以在这部分里指定这些选项。如果一个或多个选项被选择，如果活动窗口类型被指定，菜单就会被激活。

16.编辑器

允许你编辑一些 SQL 或 PL/SQL 源的每一个窗口都使用同一个编辑器。这个编辑器有很多能使程序员工作起来更轻松的特性。

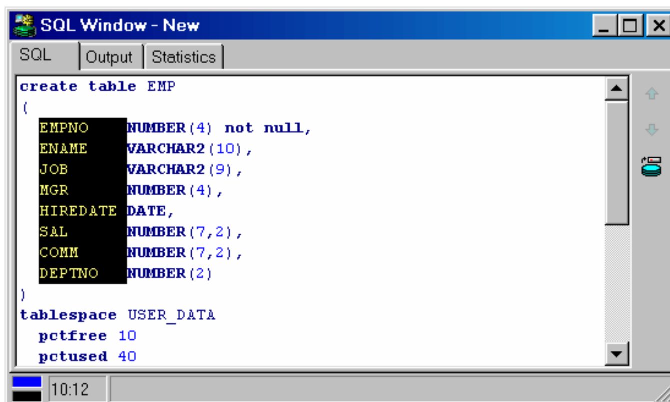
16.1 选择功能

编辑器允许你在已选的文本里执行各种功能，这些可用的功能来自于*编辑*菜单里的*选择*项目，只适用于你已经选择了的文本。你可以通过键配置首选项给这些功能分配一个功能键。

- **缩进 / 撤消缩进**
  缩进字符的数量依赖于编辑器首选项。如果文本的选择存在，*Tab* 和 *Shift-Tab* 键总是被指派到这两个功能上。
- **小写 / 大写**
- **注释 / 撤消注释**
  使用 /* 和 */ 可以快速地注释一个选择。
- **应用语法大小写**
  如果你已经把编辑器关键词大小写设置为大写、小写或首字母大写，那么在已选文本里这个功能将应用这种风格到所有的关键词。
- **排序**
  选择的行将按升序来排序。如果你再一次调用排序功能，选择将按降序来排序。

16.2 列选择

当你做一个选择时，通过按下 *Alt* 键，你就可以选择一个文本列了：



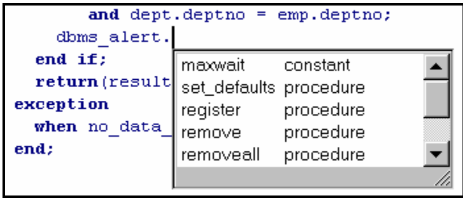
你通常可以对这个选择进行复制、粘贴、剪切、删除、大写、小写、（撤消）缩进和（撤消）注释等操作。如果你使用应用语法大小写功能并且关键词横过列边，它将不被处理。

16.3 指引线

当编辑带有多重 if/then/else 语句和跨过几页的 PL/SQL 代码的循环这样复杂的代码时，在编辑器里有一条垂直的指引线可能是有用的。要做到这一步，仅仅把光标放到你希望的水平位置并选择来自于*编辑*菜单的*指引线*项目就可以了。这个菜单项目起一个拴牢作用，所以你可以再一次使用它来移除指引线。

16.4 代码助手

代码助手是一个很强大的工具，在你键入 SQL 或 PL/SQL 代码之时为你提供帮助。例如，当你键入 *dbms_alert.* 并且稍微犹豫一小会儿时，编辑器将显示出 *dbms_alert* 包里的元素列表：



当你继续键入 *dbms_alert* 过程的名称时，列表中的内容将被简化。例如，如果你在 *dbms_alert.* 后面键入一个 *r* ，那么就只有 *register* 、*remove* 和 *removeall* 过程将仍然出现在列表里。当你在代码助手可见时按回车键时，当前选择的项目名称就被插入到源里了。你还可以在代码助手窗口里使用箭头键来选择需要的项目。按 *ESC* 键将移除代码助手。

代码助手将为下列对象类型提供帮助：

对象类型	元素
包	过程、函数、类型、变量、常数和异常
函数、过程	参数、带有指定符号的插入（param => ）
表、视图	列
序列	nextval 和 currval
用户	被用户拥有的对象
游标变量	游标的 select 语句的字段
记录类型变量	记录类型的字段
表%rowtype 变量	表的列
对象类型	属性（对于默认构造器）
对象类型变量	属性和方法
对象类型列	属性和方法
集合变量	集合方法（first，next，limit，等等）

代码助手还可以帮你键入对象的名称（表、包等等）、关键词和在当前前后关系内部有意义的 PL/SQL 标识符。例如，如果你在 SQL 编辑器里键入 *pro* 并犹豫一下，关键词 *procedure* 和 *profile* 将被代码助手显示出来，连同开头为 *pro* 的任何数据库对象名一起。前后关系依赖于当前连接的用户（它决定了可用的对象名）、编辑器类型（它决定了可用的关键词）和你要编辑的程序单元（它决定了可用的 PL/SQL 标识符）。

代码助手可以自动/手动地被调用，在键入一个对象名后代码助手被自动激活前的延迟时间可以通过首选项来定义。此外，你可以定义你是否要描述一个用户，如果是，你要求被包括进去哪些对象类型。前后关系描述和需要被键入的字符的最小数量在它被激活前还是配置好了为好。所有这些首选项都在第 14.13 章里有描述。

16.5 重新调用语句

无论你何时成功地在 SQL 窗口、命令窗口或报告窗口里执行了一个语句，这个语句都被储存到全局语句重新调用缓冲区里。通过选择来自于*编辑*菜单的*重新调用语句*项目，或者通过按 *Ctrl-E* 按钮，你可以在编辑器里重新调用最近执行的语句。这将调出一个象这样的选择列表：

Recall Statement

Time	User	Statement
28-10-2001 14:29:16	scott	select * from emp
28-10-2001 14:29:09	scott	update emp set sal = round(sal)
28-10-2001 14:28:43	scott	update emp set sal = round(sal, 2)
28-10-2001 14:27:35	scott	select * from emp t where t.deptno = 10
28-10-2001 14:27:29	scott	select * from emp t where t.empno is null
28-10-2001 14:26:47	scott	select * from dept t
28-10-2001 14:25:47	sys	select * from dba_role_privs p where p.grantee= 'PUBLIC'
28-10-2001 14:25:28	sys	select * from dba_role_privs p where p.grantee = 'SCOTT'
28-10-2001 14:24:39	scott	grant select, references on emp to public
28-10-2001 14:24:19	scott	analyze table dept compute statistics

语句按执行顺序显示，最近一个首先显示。在列表里，你可以看到时间、用户和（部分）语句文本。现在，你可以选择一个语句，双击它（或按回车键）把它插入到列表被调用的编辑器里光标所在的位置。

注意，由于安全的原因，包含口令的语句不会被放到全局语句重新调用缓冲区里。

通过按列表标题里相应的按钮，你可以按时间、用户或语句文本的顺序来给列表排序。要只显示被当前连接的用户执行的语句，请右击列表并选择来自于弹出式菜单的*当前用户*项目。在同样的弹出式菜单里，你可以转回到*所有用户*，你还可以删除语句或复制语句文本到剪贴板上。

要查找一个特定的语句，通过使用相应的功能键（默认为 *Ctrl-F*），或者通过右击列表并选择来自于弹出式菜单的*查找语句*项目，你可以调用查找功能。查找功能将在当前已选语句的后面开始执行，若没有找到就在第一个语句处继续进行。

此外，弹出式菜单还包含了这些功能：从列表里删除已选语句、复制语句到剪贴板或导出整个列表到 CSV 文件。

重新调用缓冲区可以容纳 200 个语句，这些语句都将被储存到你的 Windows 概要文件的应用数据目录里。通过设置首选项，你可以改变这个声明的限制和目录（请看第 14.24 章）。

16.6 专用复制

如果你正在用 PL/SQL Developer 写 SQL 和 PL/SQL 代码，随后你又要在其它工具里使用代码，例如象 3GL 这样的程序设计语言，那么你可能需要把这些代码转换为稍微不同的格式。让我们假设你已经在 PL/SQL Developer 里写了并测试了这样一个 SQL 语句：

```
select deptno, sum(sal) mgr_sal from emp
where job = 'MANAGER'
group by deptno
order by mgr_sal desc
```

例如，如果你要在 Borland Delphi 里使用这个语句，你可能需要象这样的格式：

```
SQL := 'select deptno, sum(sal) mgr_sal from emp' + #13#10 +
       'where job = ''MANAGER''' + #13#10 +
       'group by deptno' + #13#10 +
       'order by mgr_sal desc';
```

为了这个目的，在右击了编辑器里的一个选择后，你可以使用来自于*编辑*菜单或来自于弹出式菜单的*专用复制*功能。这个功能有一个子菜单，它显示了所有被定义的专用复制格式。在选择了格式之后，被转换的代码就被储存在剪贴板上了，这样你就可以粘贴它到相应工具的编辑器里了。

专用复制格式被定义在 PL/SQL Developer 安装目录下的 SpecialCopy 子目录里。你可以改变预先确定的复制格式或者添加新的复制格式。仅仅简单地添加一个带有 .copy 扩展名的文本文件就可以了，它包含了一个针对 PL/SQL 代码第一行的变量（<line_1>）、一个针对 PL/SQL 代码最后一行的变量（<line_N>）和一个针对所有其它行的变量（<line_*>）。下面是一个针对 Borland Delphi 的例子：

```
;PL/SQL Developer SpecialCopy definition for Borland Delphi
;<line_1> for first line
;<line_*> for all other lines
;<line_N> for last line
;
SQL := '<line_1>' + #13#10 +
       '<line_*>' + #13#10 +
       '<line_n>';
```

第一行需要为指派到 SQL 的变量加上前言，接下来需要有一个 CR/LF 对。最后一行不需要有 CR/LF 对，但需要用分号来终止。所有其它行仅仅需要 CR/LF 接在后面。

如果 <line_1> 和 <line_n> 都与 <line_*> 一样，你可以忽略它们。

在一些语言里，你需要对特定的字符使用换码序列。例如，在 C++ 里，你要对 tab 字符（ASCII 码为 9）使用 \t 。要定义这些换码序列，请使用 #define 关键词：

```
#define char(9) = \t
#define \ = \\
String("<line_1>\n") +
String("<line_*>\n") +
String("<line_n>");
```

你还可以使用“#define compress”来指出你要从结果里移除所有多余的空字符（空格、制表符和换行）。

注意，.copy 文件的名字将被包括在菜单里，所以你应该使用描述性的文件名。

16.7 前后关系敏感帮助

大多数程序员经常访问他们正在程序里使用的功能的参考信息。要尽可能快地找到这些参考信息，请把光标放在源文本里的关键词上面，再按 *F1* 就能搜索 Oracle 在线手册里的这些关键词。

要了解关于 PL/SQL Developer 帮助系统的更多信息，你可以阅读第 23 章。

16.8 数据库对象弹出式菜单

当你的程序访问了一个数据库对象时，你经常需要从这个对象里获得一些信息，或者你需要处理它。你可能需要知道它的描述或它的属性。对于表或视图，你可能需要查询或编辑数据；对于函数或过程，你或许需要测试某一设想。

如果你右击编辑器里的对象名，你将能看到与你右击对象浏览器里的对象名时看到的一样的弹出式菜单。因此，所有针对这个对象的相关功能将在编辑器里立即可用。

16.9 解释计划

要为编辑器里的 SQL 语句调用解释计划窗口，请选择语句的文本，右击它，再选择来自于弹出式菜单的*解释计划*项目。如果没有文本被选择，整个源都会被带到解释计划窗口里。如果 SQL 语句包含了 PL/SQL 变量，你必须在解释计划窗口里改变语句，就象第 5.1 章里描述的那样。

16.10 宏

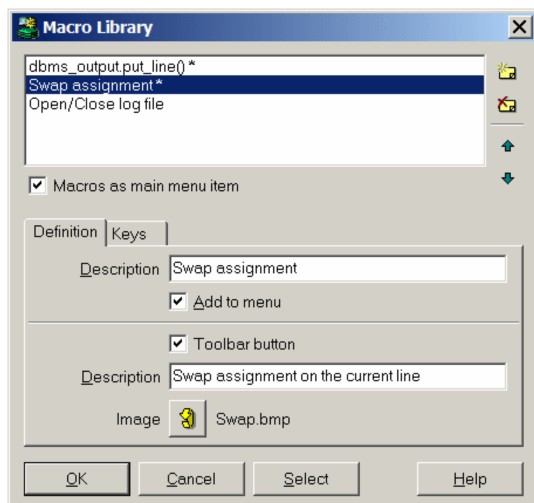
编辑器有一个强大的宏功能，允许进行快速和容易的宏记录和宏回放，这能被用于自动执行一个特定的重复的任务。

被记录的宏能被存储于宏库里，以便以后运行。库里的宏通常将是一些反复执行的重复任务（例如交换赋值：a := b; -> b := a;），或者可能包含一些代码片断。例如，当你可以一次性地键入游标类型并可以记录下来留着以后使用时，为什么还要不断地查寻一个游标类型应该怎样被声明呢？

要记录一个宏，请按 *F11* 或选择*工具*菜单里的*宏*菜单里的*记录*项目。现在，工具栏里显示出了一个闪烁的记录器图标，它表示你当前正在记录一个宏。所有的键盘点击

都被记录下来留待以后回放，这意味着在宏被记录期间你不应该使用鼠标。再一次按 *F11* ，宏记录就被终止了。按 *F12* 将回放宏。

要在宏库里储存当前记录的宏，请选择*宏*菜单里的*库*项目。这将打开宏库对话框：



按*新建*按钮可以添加一个带有当前记录定义的新的宏。你可以为宏输入描述并指出它是否应该被包括到宏菜单里。这样你可以容易地生成一些可用的宏，并且你还可以通过键配置首选项给每一个宏指派一个功能键。如果*作为主菜单项目的宏*选项被检验，那么*宏*菜单就被定位到主菜单里。如果没有被检验，它将位于*工具*菜单的下面。

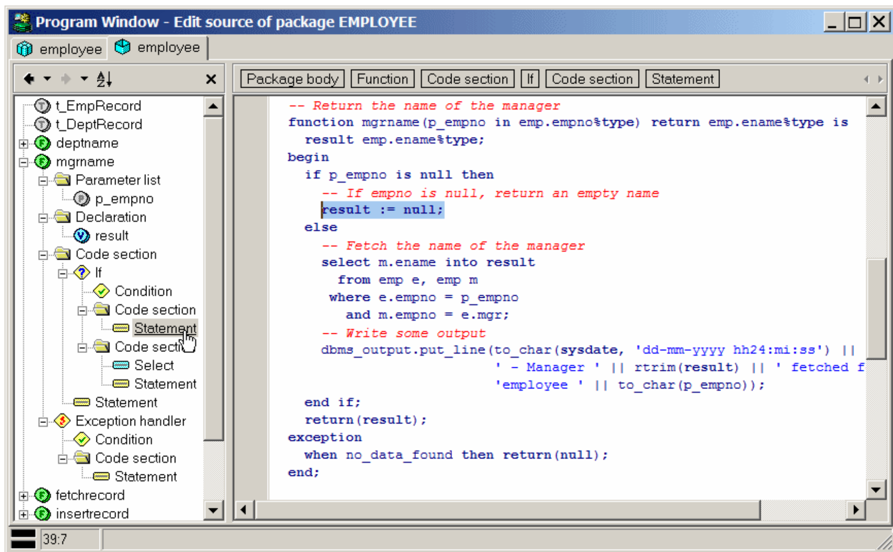
当*工具栏按钮*选项被允许时，外部工具还可以被包括到工具栏里。当你控制鼠标光标越过工具栏按钮时，相应的*描述*将被作为提示显示出来。如果你保留这个描述为空，则主描述（与显示在库里和在菜单里的一样）将被使用。按*图象*按钮来为工具栏按钮选择一个 Windows 位图文件（*.bmp）。这个图象的大小应该是 20 x 20 像素比较合适。注意，PL/SQL Developer 将始终从原始位置载入位图文件，因此若相应的宏不改变你就不要移除或重新命名这个文件。PL/SQL Developer 带有许多标准的位图文件，你可以从那里挑选它们。这些文件位于 PL/SQL Developer 安装目录里的 Icons 子目录里。这是图片选择器默认的目录。

在*键*标签页上，你可以为宏查看和改变键盘的作用。

要执行一个已存储的宏，请打开宏库并双击它的名称。现在，已选的宏可以通过按 *F12* 被执行了。当你在菜单里选择宏时，它将立即被执行，并且可以通过按 *F12* 重新被执行。

16.11 代码目录

代码目录特性被限制到程序编辑器里，它显示了程序单元的完整结构。这在大的包体和类型体里是最有用的，它可以跨越数以千计的 PL/SQL 代码行。代码目录被显示在程序窗口左边的目录树里：



代码目录明了你在源代码里的当前位置。当你操作经过源代码时，已选项目将指出你的准确位置。

如果在代码目录里你在一个项目上面移动鼠标，编辑器里相应的源代码将象上面的插图里的一样被高亮显示。

在代码目录里点击一个项目将立即导向它的声明。双击项目将展开或折叠它。右击项目将显示出一个弹出式菜单，它允许你在编辑器里选择相应的源代码，允许你剪切或复制它到剪贴板上，允许你从剪贴板里粘贴它，允许你把它转换为一个注释，允许你展开所有的子项目，或者允许你为它创建一个测试脚本（如果它是公共的程序单元的话）。

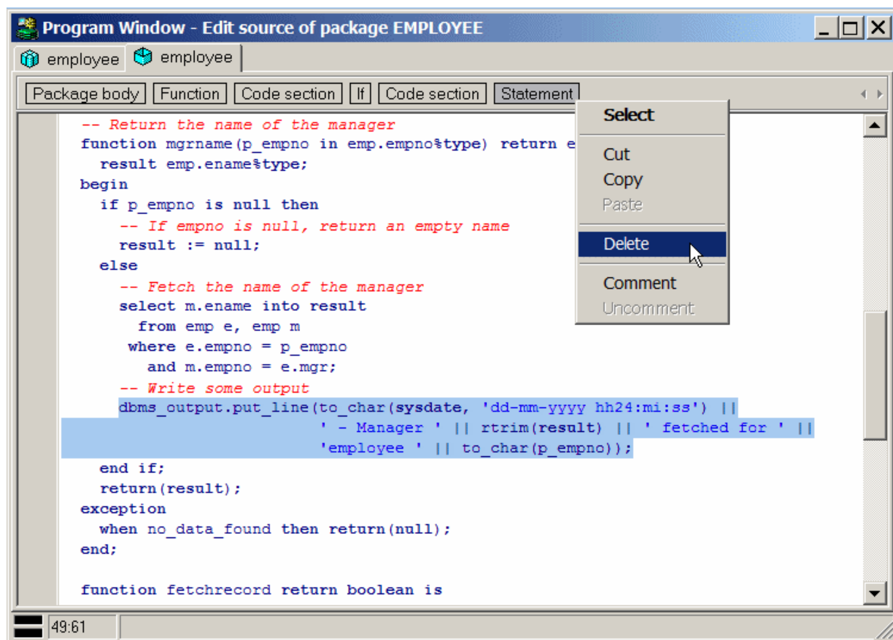
通过按目录窗格顶部的*排序目录*按钮，目录可以按字母顺序被排序。仅仅主目录将被排序，参数、本地变量等等将不被排序。按钮将蹲伏在那里，再按一次按钮将把目录恢复到它们的原始顺序。

要关闭代码目录，你可以点击右上角的*关闭*按钮。你还可以在编辑器里右击并选择*目录*项目来使它隐藏或显示，或者选择*工具*菜单里的*代码目录*项目。

在默认的情况下，代码目录是可见的。要改变这个默认，你可以关闭当前程序窗口里的代码目录并且选择*窗口*菜单里的*保存版面*项目。

16.12 代码层次

代码层次被限制到程序编辑器里，并在光标位置显示 PL/SQL 代码的层次结构。它位于编辑器的上方：



在上面的例子里，光标被定位于一个包体内的函数内的代码部分内的 If 语句内的代码部分内的 *dbms_output* 语句上。如果你在代码层次的项目上面移动鼠标，那么相应的源代码将被高亮显示。点击一个项目将选择相应的源代码。

如果你右击一个项目（与上面的插图里的一样），一个弹出式菜单将被显示出来，它允许你选择源代码、剪切或复制源代码到剪贴板上、在项目上粘贴剪贴板里的内容、删除源代码、转换源代码到注释或者把注释转换为源代码。

16.13 超链接导航

在你的 PL/SQL 代码里，你常需要在别处定义的元素做参考。例如，你可以参考同样的包体内、或同样的包的说明内、或另一个包内的一个 PL/SQL 类型，去找同样的函数或过程调用、变量、视图等等。

要从一个程序窗口内容易地找到一个声明，你可以在这样一个参考上面移动鼠标指针，按下 Ctrl 键，再按鼠标左按钮。在按下 Ctrl 键后，在鼠标指针下面的标识符将变为淡蓝和有下划线，类似于 HTML 文档里的我们熟悉的超链接：

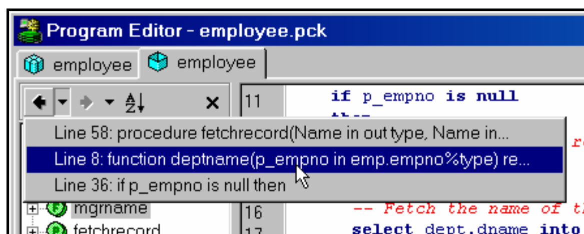
```
-- Created : 17-12-1999 17:19:26
-- Purpose : Lock an employee record
procedure lockrecord(p_empno in emp.empno%type) is
  dummy varchar2(1);
begin
  if p_empno != g_lastrecord.empno then
    select null into dummy from emp
    where empno = p_empno
    for update;
  end if;
end lockrecord;
```

如果声明位于同样的源内， PL/SQL Developer 将导向那里。在那之后，同样的程序窗口内的剩余的程序单元将被搜索。然后，它将搜索所有其它的当前打开的程序窗口。最后，它将为包含声明和将导向那里的对象设法打开一个新的程序窗口或 SQL 窗口。

在默认的情况下，如果适当的话，超链接导航将在包或类型说明里定位一个声明。要打开包或类型体以取代说明，你可以右击超链接。

16.14 浏览按钮

无论你何时跳到程序窗口里的一个绝对位置， PL/SQL Developer 都将为早先的位置生成一个书签。这些书签可以用向后浏览和向前浏览按钮而被访问：

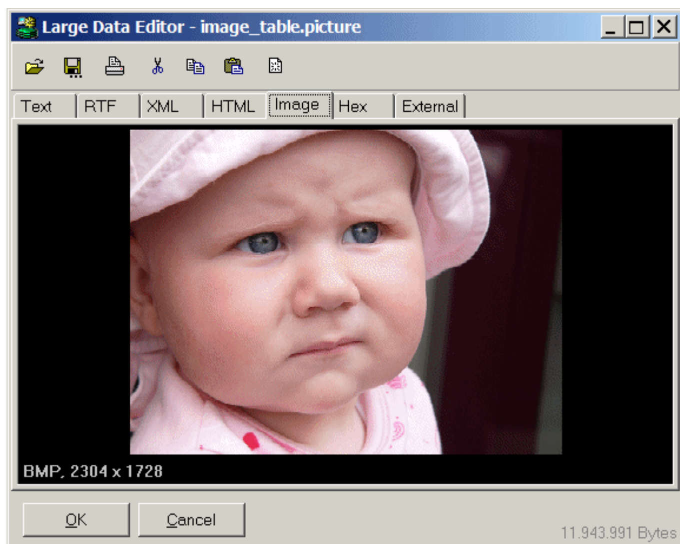


当你在代码目录里航行，使用查找功能，点击一个编译错误，转到编辑器的顶部或底部等等的时候，书签将被添加。

17. 大数据编辑器

只要你需要查看或编辑 long 、long raw 、CLOB 、BLOB 或 BFILE 列或者变量值，大数据编辑器都将被调用。它还能被用于 varchar2 数据。大数据编辑器将分析数据并尝试在正确的格式里显示它。它可以显示的值如文本、RTF 、XML 、HTML 、图像（bmp 、jpg 、gif 、tiff 和其它格式）和十六进制数据。此外，它还可以运行外部应用程序（例如 MS Word 或 Acrobat Reader）来查看或编辑数据。

为图像调用大数据编辑器将得出下列屏幕里的结果：



编辑器顶部的工具栏包括了下列按钮：

- 打开 – 打开文件内容到编辑器里。
- 保存 – 保存编辑器内容到一个文件里。
- 打印 – 按当前已选格式打印编辑器里的内容。
- 剪切 – 按当前已选格式把编辑器里的内容放到剪贴板上，随后清除（null）编辑器里的内容。
- 复制 – 按当前已选格式把编辑器里的内容放到剪贴板上。
- 粘贴 – 按当前已选格式把剪贴板上的内容粘贴到编辑器里。
- 清除 – 清除（null）编辑器里的内容。

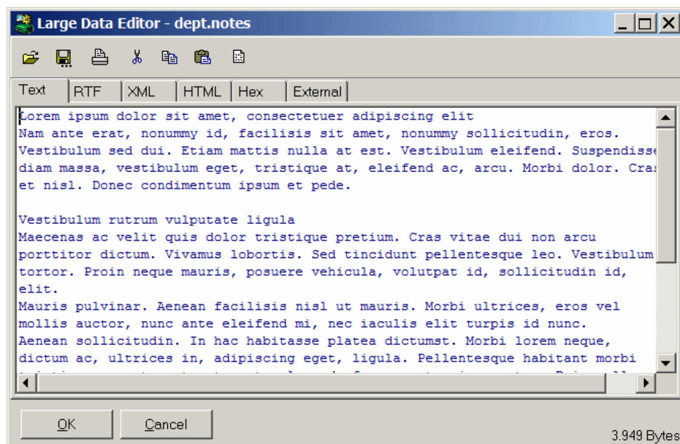
在工具栏的下面，你可以为编辑器里内容的正确格式选择标签页。对于一些数据，有多种格式是可以用的。例如，HTML 文档可以以 HTML 格式或以文本格式被查看。所有工具都可以按十六进制格式（二进制格式）被查看。

当特定格式的编辑器内容发生改变时，你可以立即转到一个不同的格式来查看这些改变。例如，当查看一个 HTML 文档时，你可以转到文本格式来产生改变，再转到 HTML 格式来查看这些改变。

在按了确定按钮之后，你的改变将被传递到来自大数据编辑器被调用的地方的功能里。

17.1 编辑纯文本

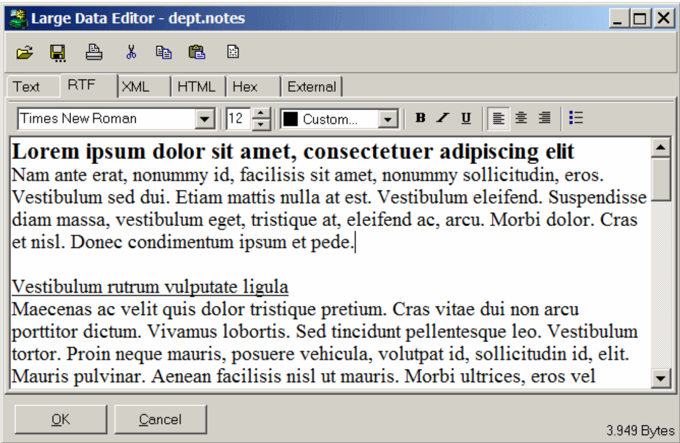
文本编辑器可以被用于查看或编辑任何种类纯文本数据：



它还可以被用于编辑 XML 和 HTML 文本。转到 XML 或 HTML 标签页将立即显示这些改变的结果。

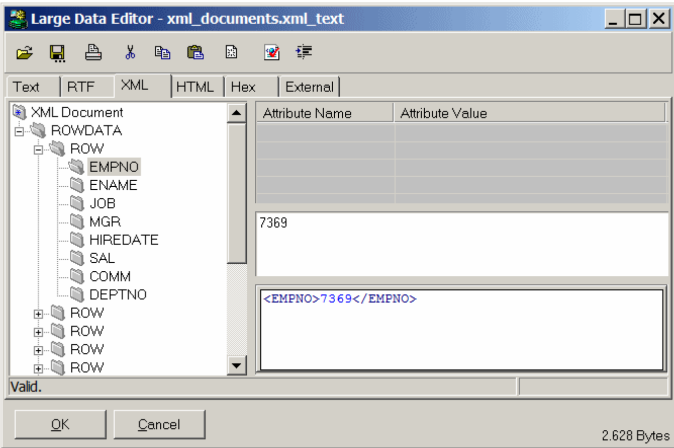
17.2 编辑 RTF

RTF 编辑器可以被用于查看和编辑富文本格式的文本。它允许你容易地改变字体类型、大小、颜色、风格和对齐：



17.3 编辑 XML

当查看 XML 内容时，编辑器首先转到 XML 标签页：

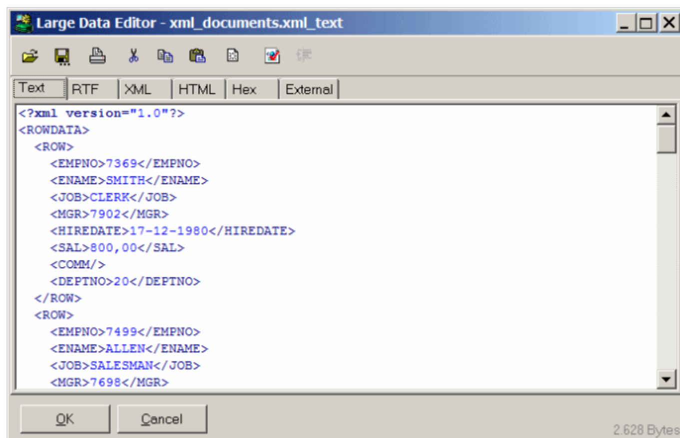


在这里，你可以看到树形视图里的 XML 文档的结构，可以编辑各个元素和属性。

按*解析 XML* 按钮将解析 XML 文档，并将报告任何错误。这需要你有 Net8 8.1 或更新版本，并且用户可以访问到 SYS.XMLTYPE 类型或 XMLPARSER 包。

你可以按工具栏里的*格式化 XML* 按钮在文档里应用标准的格式标准。

如果你为 XML 文档转到纯文本格式的标签页，那么适当的语法高亮将被应用。



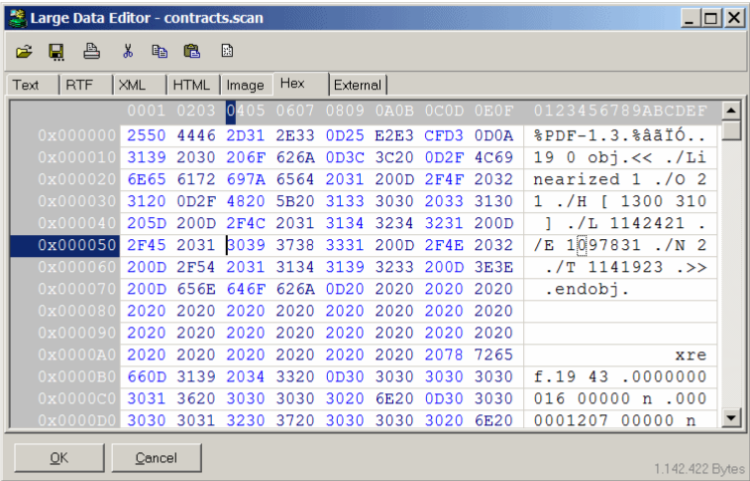
17.4 编辑图象

图象编辑器将显示最流行格式的图象，例如 JPG 、BMP 、GIF 等等。如果图象比编辑器能显示的图象大，那么这个图象将被调整大小以适应编辑器。这不会影响图象的内容，因为它仅仅是一个视觉效应。在图象的下面，你可以看到图象的格式和图象的像素大小。

要改变一个图象，你既可以按工具栏里的*打开*按钮从文件里调用它，又可以从另外一个源复制图象到剪贴板上并按*粘贴*按钮。注意，当经由剪贴板复制/粘贴图象时，图象格式将改变到 BMP 格式！要保证原始格式被保持住，你应该从一个文件里载入它。

17.5 编辑十六进制数据

要查看或编辑编辑器里十六进制格式的内容，你可以转到十六进制标签页：

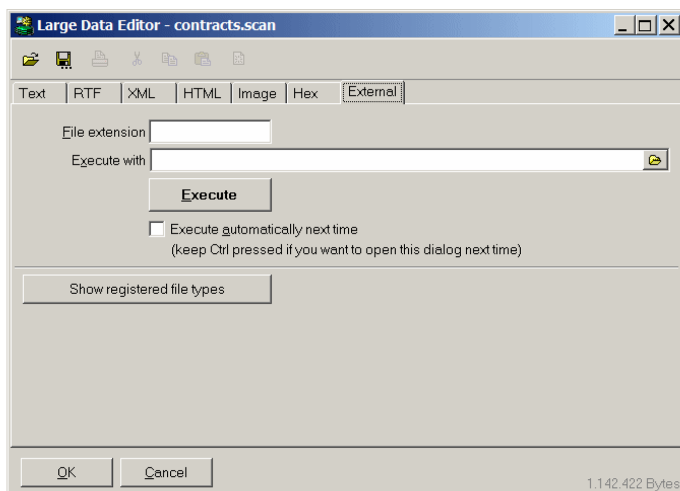


在左边，你可以看到带有十六进制数据的词（16 位）的 8 个列。在右边，你可以看到十六进制数据的文本内容。你既可以编辑十六进制的内容又可以编辑文本的内容。

如果你右击十六进制编辑器，那么你可以选择一个不同的列的版面。

17.6 调用外部查看器或编辑器

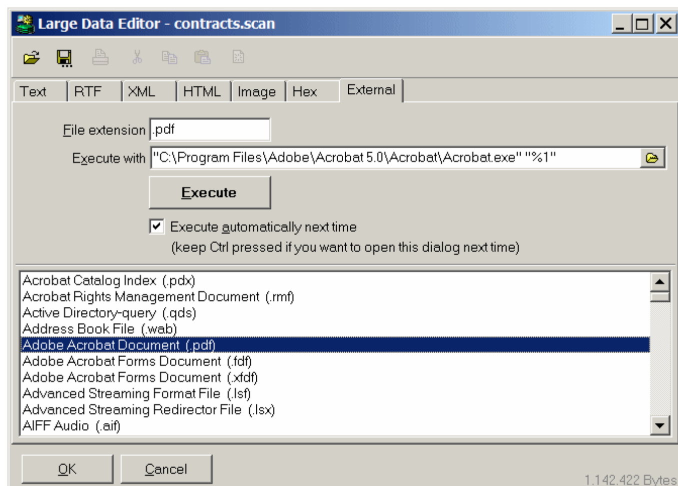
对于很多被储存在数据库里的大对象，调用外部查看器或编辑器可能是必要的。例如，如果一个 BLOB 列包含了 Acrobat PDF 文件（轻便文档文件）的内容，它将必须调用 Acrobat Reader 。当为这样的内容调用了大数据编辑器之后，你将首先被带到文本或十六进制标签页。在那之后，你可以转到外部标签页：



在这里，你可以定义哪一个程序应该被执行。首先，你需要定义文件的扩展名。在执行外部程序之前，带有编辑器内容的临时文件将被带有那个扩展名来保存。其次，你需要定义哪一个程序应该被执行。在那之后，你可以按*执行*按钮来查看外部程序里的内容。如果你在外部程序里改变了文件并保存了它，大数据编辑器将注意到这些改变并且把这些改变导回到编辑器里。

当查看列数据时，表/列与外部程序二者的关系将被记住，以便你下次查看它。另外，如果你激活*下次自动执行*选项，那么大数据编辑器将立即调用外部工具。要越过这个自动行为，当调用编辑器时按下 *Ctrl* 键就可以了。

大多数程序已经注册了它们的文件类型，以取代手工定义扩展名和程序，你可以简单地从列表里选择它。首先你需要按*显示已注册文件类型*按钮，然后你可以搜索正确的文件类型并双击它。对于 PDF 文件，这将是 *Adobe Acrobat 文件*：



如果你右击文件类型列表，你可以按扩展名或描述（默认）的顺序来排序。

18. 查询设计器

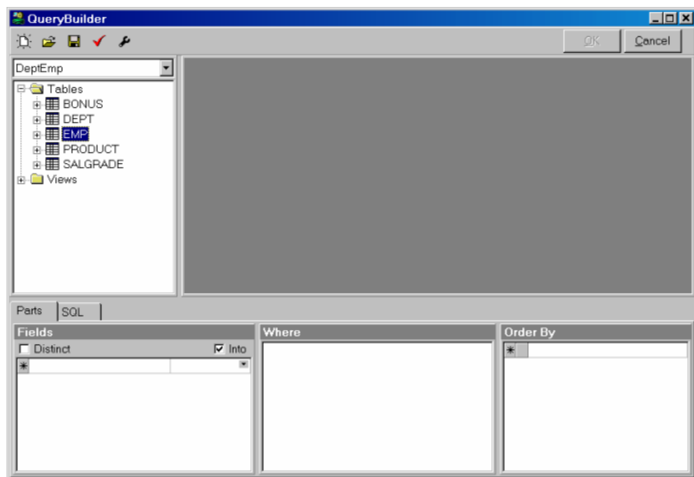
查询设计器允许你创建和修改你的 PL/SQL 和 SQL 源文件里的 select 语句。它提供了简单的指和点界面来建立 select 语句的字段列表、表列表、where 子句和 order by 子句。

18.1 创建新的 select 语句

对于这个例子，我们将假定我们要在 dept 和 emp 表之间创建一个连接语句，并显示一些来自于这两个表的一些列：

```
select e.empno, e.ename, d.deptno, d.dname
from emp e, dept d
order by e.empno
```

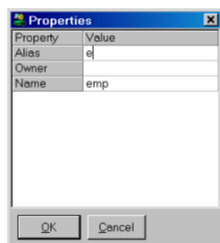
要创建一个新的 select 语句，请把编辑器指针放到你要它被插入的文本位置，再按工具栏里的*查询设计器*按钮（或选择来自于弹出式菜单的*查询设计器*项目）。这将调出一个空的查询设计器表单：



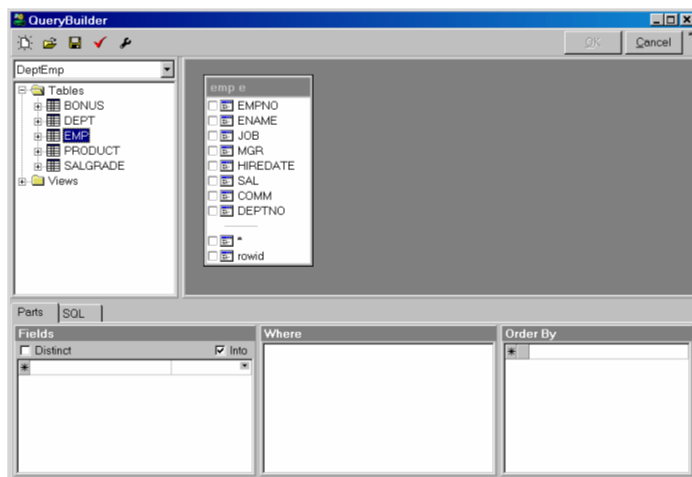
在左边，你可以看到一个只带有表和视图的对象浏览器。在右边，你可以看到工作区，现在正好是空的。在底部，你可以看到带有字段列表（*into* 项目列表可选）、where 子句和 order by 子句的三个面板。底部包含了一个标签页，它允许你转到将从当前查询定义里被创建的实际的 SQL 文本。

在窗口的顶部，你可以看到一个工具栏，它包括了创建新查询的*新建*按钮、打开来自于文件的以前保存的查询定义的*打开*按钮、保存当前查询定义到文件里的*保存*按钮、解析当前查询定义的*解析*按钮和*首选项*按钮。*确定*按钮把你转回到 PL/SQL 或 SQL 编辑器，并且插入你已经建立的 SQL 文本。

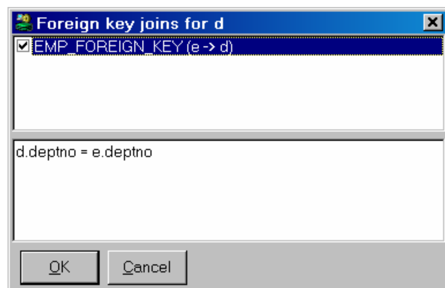
要在 select 语句里包括表或视图，请把它们从对象浏览器里拖出来放到工作区里。你每次添加一个新的表，*表属性*对话框都将显示出来：



在这里，你可以输入表的别名。对于我们的例子，我们要使用‘e’作为 *emp* 表的别名：

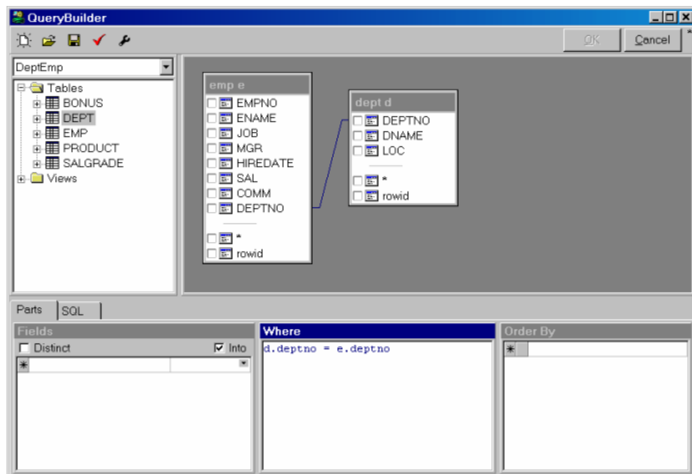


在用同样的方法添加了 *dept* 表之后，你会被问到是否使用名为 *EMP_FOREIGN_KEY* 的外键列来连接两个表：

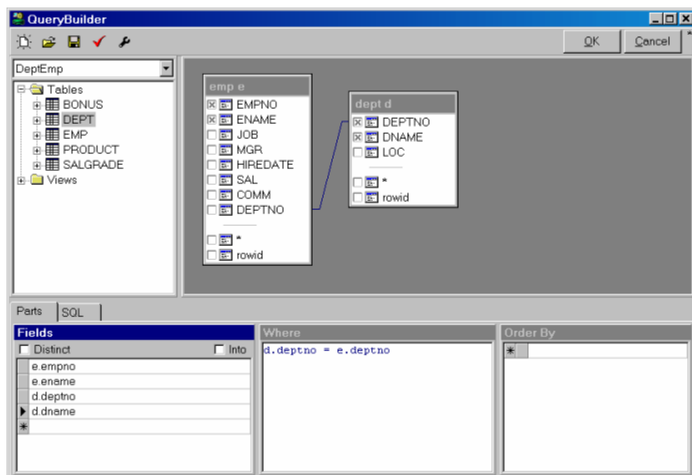


在检验了这个选项之后，你将在表单的下半部看到新的连接条件。

现在，*Dept* 表被添加了，连接条件在新的工作区里被显现出来了：

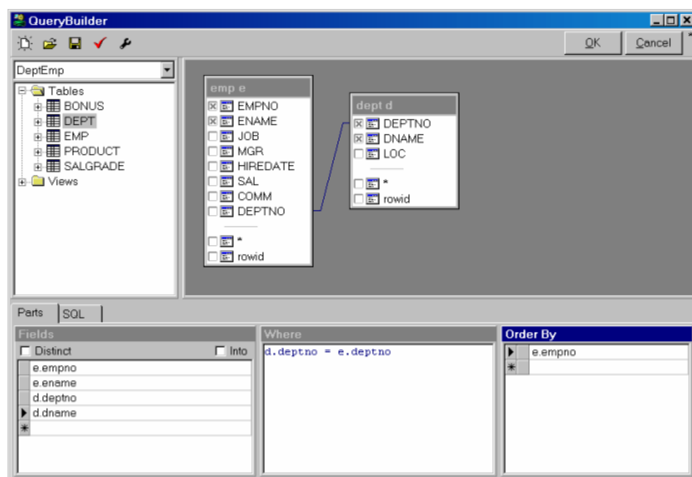


现在，我们要做的事情是在 select 语句的字段列表里加入列，只要点击列名旁边的检验栏就可以了。列将被显示在*字段*面板里：



如果你要创建一个 PL/SQL *select .. into* 语句，你还必须为邻近字段的 *into* 列表指定项目。这个选择列表将显示出在你插入语句的 PL/SQL 代码里被定义的 PL/SQL 变量。在这样的情况下，我们不需要创建 *into* 列表，所以我们可以保留这些项目为空。

要创建 order by 子句，把 *empno* 列拖到 *Order By* 面板里即可：



现在，你可以按*确定*按钮来把 SQL 语句插入到编辑器里了。

18.2 修改现有的 select 语句

要在你的 PL/SQL 或 SQL 源代码里修改现有的 select 语句，简单地右击 select 语句并选择来自于弹出式菜单的*查询设计器*项目即可。整个 select 语句都会在编辑器里被标记，并且查询设计器将被显示出来。现在，你可以象前一章描述的那样来改变查询定义了。

18.3 处理查询定义

下列的段落描述了你可以怎样处理查询定义。

改变字段列表

要从工作区的表里添加字段到查询，你可以点击列旁边的检验栏，或者你可以把列拖到字段列表里。作为一种选择，如果字段列表拥有焦点，你还可以双击列。

你还可以直接把列或列文件夹从对象浏览器拖到字段列表里。列文件夹位于表、它的约束和它的索引的下方。

你可以直接在字段列表里键入列名和其它表达式。

要从字段列表里删除一个字段，请右击它并选择来自于弹出式菜单的*删除已选项目*。如果你选择了多个字段，你可以一下子把它们都移除。

通过选择它们，你可以把字段移动到不同的位置，把它们拖到一个新的位置。

如果你要一个 *distinct* 结果集，在字段列表上方检验相应的选项就可以了。

使用字段别名

要使用字段别名，只要在表达式后面键入别名就可以了，就象你正在用手键入 select 语句一样。

改变 where 子句

Where 子句是标准的文本编辑器，在这里你可以输入条件。要包括一个列，你可以从工作区的表里拖它，或者 where 子句若有焦点你可以双击列。

要添加一个基于外键约束定义的连接条件，请右击工作区里的表并选择来自于弹出式菜单的*外键*项目。你不能使用这个功能来移除一个现有的连接条件，必须用在编辑器里移除文本以代之。

改变 order by 子句

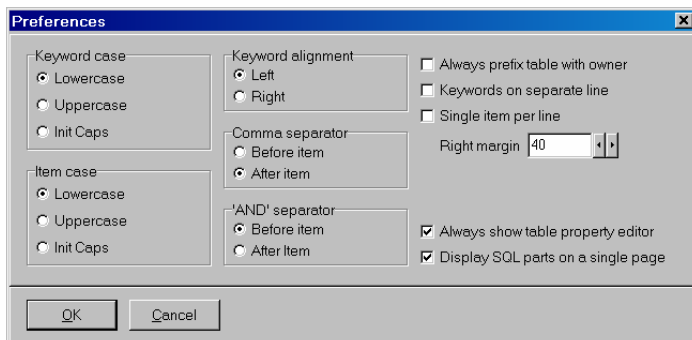
Order by 子句可以用与字段列表同样的方法来处理，除了你不能使用列名旁边的检验栏之外。要在升序和降序之间改变一个项目，你可以点击项目旁边的按钮，它将显示出一个相应的箭头。

使用表或视图的同义词

要通过同义词使用表或视图，请右击工作区并选择来自于弹出式菜单的*添加表*项目。在*名称*属性里键入同义词的名称，若必要就提供一个别名，再按*确定*按钮。

18.4 查询设计器首选项

通过按工具栏里的*首选项*按钮，你可以改变查询设计器的行为。下列表单将显示出来：



这些首选项里的大多数选项影响了已生成的 select 语句的版面：

- 关键词大小写

  决定了 SQL 关键词（select 、from 、into 、where 、order by）怎样被插入到已生成的 select 语句里。

- **项目大小写**

  决定了项目（表名、列名）怎样被插入到已生成的 select 语句里。

- **关键词对齐**

  决定了 SQL 关键词是左对齐还是右对齐。

- **逗号分隔符**

  决定了逗号是放在下一个项目前面还是放在当前项目后面。这主要用于*每行一个项目*选项被允许时。

- **'AND' 分隔符**

  决定了在 where 子句里'AND' 分隔符是放在下一个条件前面还是放在当前条件后面。

- **总是给表加上前缀所有者**

  当被允许时，表名被加上前缀所有者（例如用 scott.emp 取代 emp）。当被禁止时，如果表不被当前用户拥有，仅仅表被加上前缀所有者。

- **关键词在分隔行上**

  当被允许时，SQL 关键词被放在分隔行上，所有的项目都被放在后来的行上并用两个字符缩进。如果这个选项被禁止，项目将立即在同一行上紧跟关键词。

- **每行一个项目**

  当被允许时，每一个项目都被放到一个新的行上。当被禁止时，项目都被放到一个行上，直到到达*右边距*。

- **右边距**

  决定了在*每行一个项目*首选项被禁止时每一行的最大长度。

- **总是显示表属性编辑器**

  当这个选项被允许时，表属性编辑器将总是在添加新表到查询定义的后面被显示。当这个选项被禁止时，如果最近添加的表已经用于查询并且因此需要有一个别名，那么仅仅是属性将被显示。

- **在单一页上显示 SQL 部分**

  决定了 SQL 的三部分（字段列表、Where 子句、Order by 子句）被显示在单一的页上。如果这个选项被禁止，每一部分都被显示在一个分隔标签页上，在 SQL 文本标签页的旁边。

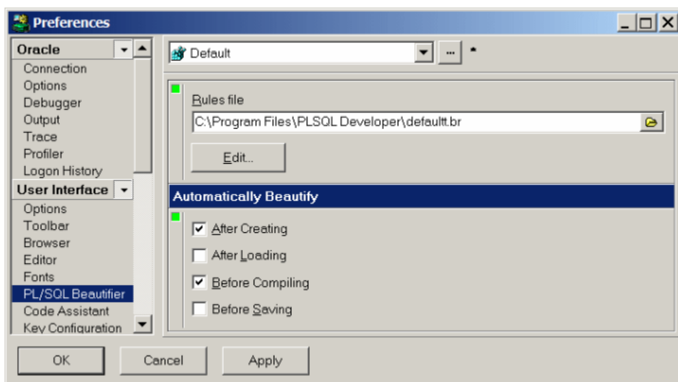
19. PL/SQL 美化器

如果你与几个开发者一起在工程小组里开发、维护和支持 PL/SQL 代码，那么所有的工程成员都使用相同的代码风格是有用的。这将对产品的质量和生产效率是有益的。要实施这个，PL/SQL Developer 提供了一个 PL/SQL 美化器。

你要做的所有事情就是设置 PL/SQL 美化器选项，并可以通过 PL/SQL 美化器规则随意地定义代码风格（你也可以简单地采用默认规则）。之后，当你工作时，你的 PL/SQL 代码将自动被美化，或者通过激活美化器选项被明确地美化。

19.1 定义选项

要定义 PL/SQL 美化器选项，请选择来自于*编辑*菜单的相应项目，下列首选项对话框将显示出来：



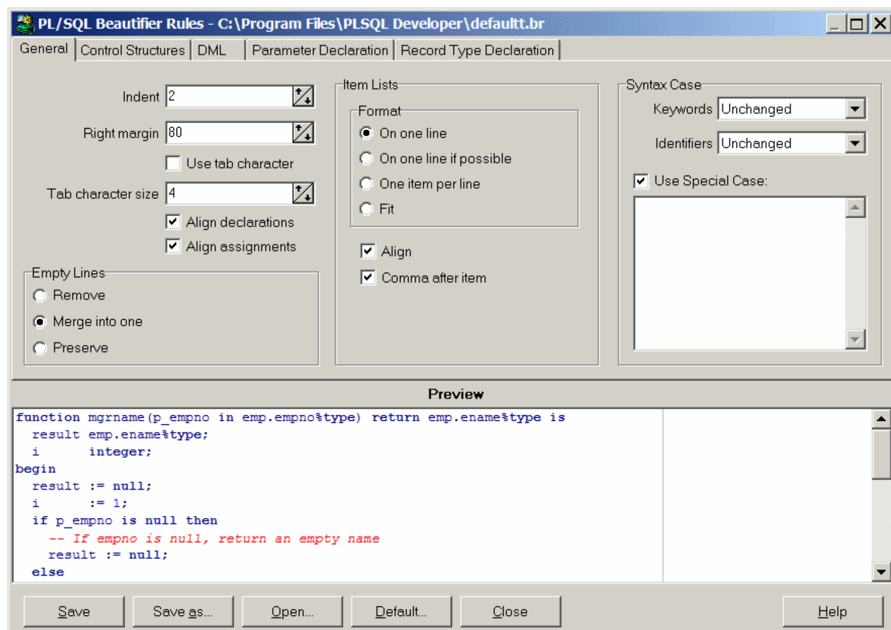
*规则文件*决定了美化你的代码时将要遵循的规则。你可以保留它为空而采用默认规则，它可以通过在 PL/SQL Developer 安装目录里的 *default.br* 文件来定义。要了解关于这些规则的更多信息，请看下一章。

此外，你还可以定义你的 PL/SQL 代码将在什么时候被自动美化：

- **在创建后** – 只要一个新的程序文件被通过模板创建，或者一个 DML 语句从对象浏览器里被拖放到程序窗口里。
- **在载入后** – 在程序文件从文件系统里被载入之后。这不影响来自于 Oracle 数据库的被打开的代码。
- **在编译前** – 当你在程序窗口里编译程序时。这保证了 Oracle 数据库里的所有 PL/SQL 代码都遵照美化规则。
- **在保存前** – 在程序文件被保存到文件系统之前。

19.2 定义规则

如果你在选项对话框里按*编辑*按钮，你就可以查看或编辑已选文件的规则：



在改变了这些规则之后，你可以按*保存*或*保存为*按钮来保存这些改变。你还可以打开另一个文件用于查看或编辑，并且通过按相应的按钮你可以恢复到默认规则。如果你以新的名称保存了文件或者打开了不同的文件，你需要在选项对话框上选择那个规则文件以使这些规则有效。

在*一般*标签页上，你可以改变应用到你的 PL/SQL 代码的一般样式的规则。你做的所有改变将立即在对话框下部的*预览*窗格里可以看到，而且大多数选项都是不说自明。下列选项需要做一些解释：

- 缩进 – 针对嵌套结构（例如 begin/end 、if/then/else 、loops 等等）将要缩进的空格数。
- 右边距 – 只要代码需要转到下一行，右边距就会被当作指引线使用。在那里也许有超越右边距的情况，例如长字符串被使用时就可能是这样。
- 使用跳格字符 – 当被允许时，作为结果的代码将允许用硬跳格符（字符 9）来缩进代码。当被禁止时，空格将被使用。
- 跳格字符大小 – 硬跳格符表现出的字符数。这在*使用跳格字符*选项被允许时并没有用，但对于正确地对齐包含硬跳格符的注释还是有用的。

- **空行** – 你可以定义空行应该被移除，还可以定义美化器将通过它自己的规则插入空行、或者后来的空行组应该合并为一个空行、或者所有的空行都被保留。
- **项目列表** – 这些规则应用到所有的不分成其它标签页类的项目列表。
- **语法大小写** – 决定了 PL/SQL 关键词和标识符怎样用大写字母来写。*使用特殊大小写*选项允许你定义指定大写的字数。例如，如果你在这个列表里包含了 *DBMS_Output* ，这将是那个词在你的 PL/SQL 代码里的准确的大写。

其它标签页包含了应用到你的 PL/SQL 代码的特殊样式的规则。这些规则还是不言自明的，被示范在预览窗格里。

19.3 使用美化器

如果你在选项对话框上还没有选择任何美化事件，通过选择*编辑*菜单的 *PL/SQL 美化器*项目、通过按工具栏里相应的按钮或通过使用被指派给它的功能键，你都可以明确地美化你的代码。在默认的情况下，在当前的 PL/SQL 编辑器里的所有代码都将被美化。如果有严重的错误出现在代码里，它就不会被美化，而且你会收到一个错误消息。

作为一种选择，你还可以选择一块完整的 PL/SQL 代码（例如，一个本地的函数或过程，或者一个 DML 语句）并激活美化器。

20. 模板

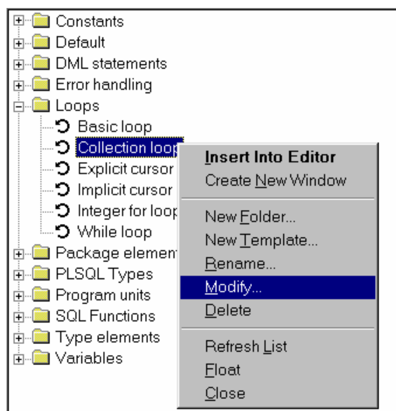
只要你创建一个新的程序单元，它的初始内容都是基于模板的，模板包含了预先定义的文本和变量。当选择了一个模板时，你将被提示输入这些当时放在模板文本里的变量的值。

模板还可以用于把 SQL 或 PL/SQL 代码片断插入到现有的源里。例如，你可以创建一个 cursor-for-loop 的模板，无论你何时需要这样的程序结构你都可以用它。

模板是纯文本文件，它位于 *Template* 目录里，以 *.tpl* 为扩展名。对于所有的程序单元类型，模板都是预先被定义好的，普通的 SQL 和 PL/SQL 结构的模板也是这样。你可以为你自己特定的环境修改和创建模板。

20.1 模板窗口

模板窗口显示出了模板目录的层次结构，这使你组织模板变得容易了。当你第一次启动 PL/SQL Developer 时，模板窗口将停靠在对象浏览器的下方。这样，它总是可用的，从不妨碍其它文档窗口的查看：



这里有很多被预先定义好的模板，它们按下列模板文件夹被组织起来：

- 常数 – 用于 PL/SQL 常数声明的模板。
- 默认 – 带有 SQL 窗口、测试窗口、命令窗口或解释计划窗口的默认内容的模板。它还包含了一个新的视图对象的模板。你大概不会在这里直接使用这些模板，但可以在这里维护它们。
- DML 语句 – 你可以在 PL/SQL 里使用的 DML 语句的模板。
- 循环 – 各种 PL/SQL 循环的模板。
- 包组件 – 能被用于包说明或包体的组件的模板。
- PLSQL 类型 – 能在 PL/SQL 里被声明的所有类型的模板。

- 程序单元 – 这是一个特殊的文件夹，它包含了针对新的程序单元的模板。只要你创建一个新的程序单元，来自于这个文件夹带有与程序单元类型一样的名字的模板就会被使用。
- SQL 函数 – 带有所有函数的模板，例如 add_months 、substr 等等。如果你不知道特定函数的准确名称或参数含义，那么这个模板就可能是有用的。
- 类型组件 – 能被用于类型说明或类型体的组件的模板。
- 变量 – PL/SQL 变量声明的模板。

在默认的情况下，模板窗口被停靠在对象浏览器的下方，但是通过右击它并选择来自于弹出式菜单的*浮动*项目你可以把它转入到浮动窗口。要再一次停靠窗口，就按同样的过程行事。

通过选择来自于弹出式菜单的*关闭*项目，你可以关闭模板窗口。要使模板窗口再现，请选择*工具*菜单里的*模板*项目。

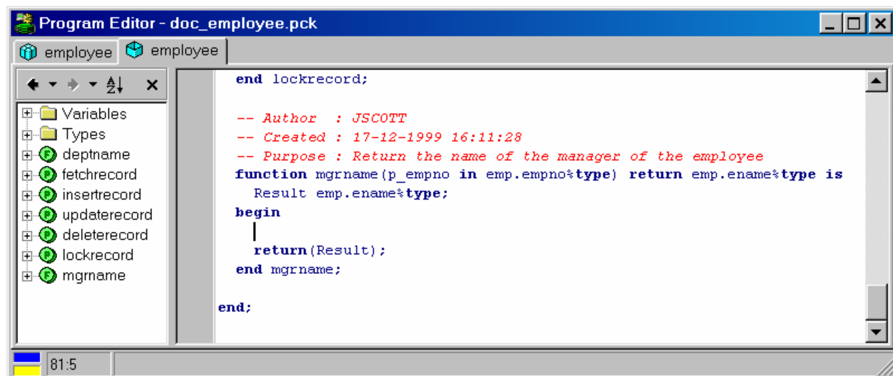
通过在*窗口*菜单里使用将保存当前状况的*保存版面*项目，你可以保存这个版面以用于你下次启动 PL/SQL Developer 。这个功能的更多详细资料在第 24.2 章里有描述。

20.2 使用模板

要插入模板到现有的源，你可以简单地双击它。如果模板包含了必须被用户指定的变量，一个输入表单将在文本被插入之前显示出来。例如，如果你双击了*函数实现*模板，下列输入表单将显示出来：

Template Wizard
Function Implementation
Purpose: Return the name of the manager of the employee
Name: mgrname
Parameters: p_empno in emp.empno%type
Return type: emp.ename%type
OK Cancel

在填了表单并按了*确定*按钮之后，模板文本将被插入到源里当前光标所在的位置：



你还可以把一个模板从模板窗口拖放到编辑器里指定的位置，这个位置也决定了模板文本的缩进。同样地，你可以右击编辑器里的一个地方，并选择一个来自于弹出式菜单的*模板*子菜单的模板项目。

20.3 创建和修改模板

要修改现有的模板，请右击它并选择来自于弹出式菜单的*修改*项目。这将调出带有模板文本的文本编辑器，这个文本包含了模板的完整说明，包括变量、查询等等。模板文本的格式在下面的章节里有描述。

要创建一个新模板，请在你要创建它的文件夹里右击它，并且选择来自于弹出式菜单的*新建模板*项目。你将首先被提示输入一个模板的名字，之后模板文本编辑器将显示出来。通过选择来自于弹出式菜单的*新建文件夹*项目，你还可以创建一个新的文件夹。

模板文本

模板说明包含了文字文本和变量。文字文本将简单地被复制到源文件里，不带有任何修改。考虑一下一个名叫 *Commit* 的模板，它只包含了文字文本 *commit;* 。当这个模板被双击时，文本 *commit;* 就被插入到光标所在位置的源里了。方括号指出了模板里的变量（见下面），所以如果你要使用方括号作为文字文本，就要使用两个方括号以代之，例如： [[option]] 。

用户变量

模板变量是模板文本里的代码，它将被用置换文本替换掉。这个置换文本可以是一个隐含的值，就象当前日期或用户名，或者它可以被用户指定。当模板被调用时，这些用户变量将在输入表单里显示出来。下面的例子为函数模板定义了变量 *Name* 和 *Type* ：

```
create or replace function [Name] return [Type] is
begin
  return(result);
end [Name];
```

当这个模板被调用时，用户可以指定 *Name* 和 *Type* 变量的值。正象你能看到的，*Name* 变量被使用了两次。用户被提示为它输入一个值只有一次，两个具体值都将被用同样的值替换。

对于用户变量，下列的功能是可用的：

- 默认值
  变量的默认值可以简单地在变量名后面被替换。要定义 varchar2 为默认函数类型，你可以使用下列声明：
  [Type = varchar2]
- 限制列表
  要为变量限制可能的值，你可以为它定义一个值的列表。要把函数类型限制到 varchar2 、数字和日期，你可以使用下列声明：
  [Type = varchar2, *number, date]
  前面带有星号的值将被用于默认。
- 建议列表
  要为用户提供一个建议值的列表，还允许变量有其它的值，请使用 … 作为列表里最后的一个值：
  [Type = varchar2, number, date, ...]
- 描述列表
  代替为列表项目和置换文本使用相同的文本，你还可以为每一个项目使用描述。下面就是带有冒号和值的描述：
  [Level = Write No Database State:WNPS, Read No Database State:RNDS, ...]
- 检验栏
  对于实际上是选项的变量，你可以使用一个检验栏。在检验栏的值之间放一个斜杠，左为未检验，右为已检验。下列变量将在检验栏为未检验时插入文本 *for each row* ，在检验栏为已检验时插入文本 *for each statement* ：
  [Statement level? = for each row / for each statement]

如果你要在置换文本里使用特殊字符（像逗号、括号等），你可以用双引号引上文本。

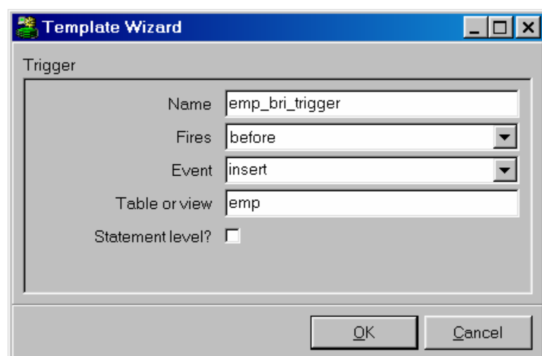
下面是一个触发器模板的例子，正好使用用户变量：

```
create or replace trigger [Name]
  [Fires = before, after, instead of] [Event = insert, update, delete, ...]
  on [Table or view]
  [Statement level? = for each row/for each statement]
declare
  -- local variables here
begin
```

```
    ;
end [Name];
```

注意，你可以在一个模板里创建多个程序单元，用只带有一个斜杠（/）字符的线来分隔它们。这样，你就可以在一个单独的程序文件里创建一个包说明和体的模板。

当这个模板被使用时，用户将被提示按下列方法输入变量值：



隐式变量

除了用户变量之外，你还可以使用隐式变量。这些变量的置换文本不被用户指定，但被系统统定义（日期、用户名），或者被模板开发者定义（查询、文本）。

系统变量

下面有四个系统变量被定义：

- $OSUSER – 操作系统用户名。
- $DBUSER – 当前登录的数据库用户名。
- $DATE – 当前日期。
- $TIME – 当前时间。

下面例子模板的前两行把 Windows 用户和当前日期/时间插入到了源文件里：

```
-- Author  : $OSUSER
-- Created : $DATE $TIME
-- Purpose : [Purpose]
procedure [Name] is
begin
    ;
end [Name];
```

注意，系统变量在模板文本里不使用方括号，但在前面加上 $ 符号。

如果你要使用系统变量的原名而不是置换值在作为结果的文本里显示出来，你可以通过在名字前放上第二个 $ 符号来避开这个置换值。例如：

```
-- $$Date$ $Revision$
```

作为结果的文本将是 $Date$ $Revision$。

光标位置

光标位置变量决定了在模板文本被插入到编辑器里之后文本光标将在编辑器里的什么地方。仅仅在你希望的地方放上 [#] 就可以了：

```
loop
  [#]
end loop;
```

查询变量

你可以使用查询在模板里组装选择列表。下列例子定义了查询 *seq_query* ，它在随后被当作 *Sequence* 变量的建议列表而被使用：

```
[$QUERY seq_query =
        select lower(object_name) from user_objects
        where object_type = 'SEQUENCE'
        order by object_name]
select [Sequence=$seq_query,...].nextval into [Variable name] from dual;
```

Sequence 变量是一个建议列表，这是因为查询被“ ,… ”跟随着，这意味着其它值能被用户用手工输入。查询结果能简单地被当作逗号分隔的值的列表来查看。

包括和排除文本

你可以在模板里包括或排除文本，这依赖于另外一个变量的值。在下面的例子里，如果 *Search condition* 被用户输入了，词 *where* 就被添加了：

```
select [Item list]
  into [Variable list]
  from [Table list]
 [+Search condition=where] [Search condition];
```

结果，用户不必在搜索条件里键入“ where ”。要排除一段文本，请使用 *–Variable name* 语法来取代 *+Variable name* 语法。

文本变量

你可以在模板里定义文本变量并在模板的其它部分里提到这些变量。如果你要有条件地包括大段的文本，这就是特别有用的。如果用户这样需要，下面的例子就包括了一个异常块：

```
[$TEXT exception_block=
exception
  when no_data_found then ...
  when too_many_rows then ...
  when others then ...
end;]
select [Item list]
  from [Table list]
  into [Variable list]
 where [Search condition];
[Exception block =/$exception_block]
```

模板图标

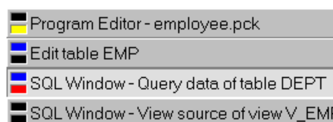
在模板窗口里树形视图为每一个模板都显示了一个图标。模板窗口首先寻找与模板名一样的位图文件名。对于模板 Cursor.tpl ，它将显示一个来自于同一个目录的名为 Cursor.bmp 的位图，如果这个位图文件存在的话。如果这个位图不存在，它将在当前目录下的目录里寻找 Cursor.bmp 文件。如果这个位图还是没有被找到，它将在当前目录里和下面的目录里寻找 default.bmp 文件。

模板图标的位图的大小应该是 16x16 像素，颜色为 16 色。

21. 窗口列表

窗口列表允许你直接和容易地在多个文档窗口之间导航。如果你喜欢用最小化文档窗口来工作，这个特性就特别有用，仅仅是你有大量的打开的窗口，这样一个列表也总是会使导航更容易。

在默认的情况下，窗口列表被停靠在对象浏览器和模板窗口的下面。它仅仅在一个或多个窗口被打开时才是可见的，并且显示了标题和这些窗口的状态：



如果你点击这些项目之一，相应的窗口就会被激活。如果你在窗口列表里右击一个项目，下面的弹出式菜单将被显示出来：



从这个弹出式菜单里，你可以激活、关闭和调整窗口大小到正常或最小状态。你还可以通过按 Shift 键并点击列表里的项目来关闭一个窗口。

此外，你可以重新命名一个窗口。如果你正好创建了一个新的窗口并且要为它提供一个有意义的名字，你又不想用比如 *SQL 窗口 – 新建*这样的名字，这就可能是有用的。

*新建*菜单项目允许你创建一个新的程序窗口、SQL 窗口、测试窗口、命令窗口或解释计划窗口。

要运行某一窗口，你可以选择*执行*菜单项目。

通过选择*浮动*项目，你可以把窗口列表转为浮动窗口。要再一次停靠窗口，请按照同样的过程来做。要关闭窗口列表，请选择*关闭*项目。要使窗口列表重现，请选择*窗口*菜单里的*列表*项目。通过在*窗口*菜单里使用将保存当前状况的*保存版面*项目，你可以保存这个版面以用于你下次启动 PL/SQL Developer 。这个功能的更多详细资料在第 24.2 章里有描述。

22. *授权*

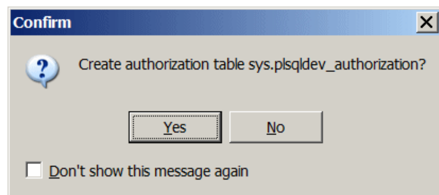
在安装了 PL/SQL Developer 之后，在系统权限和被授予到连接到数据库的 Oracle 用户的对象权限的范围内，所有的用户都能使用 PL/SQL Developer 的全部功能。例如，如果 Oracle 用户没有 *create user* 的系统权限，PL/SQL Developer 也能在 PL/SQL Developer 里启动*新建用户*功能，但最终能得到一个来自于 Oracle 的错误信息：*“ORA-01031，权限不够”*。

你可以明确地把所有相关的 PL/SQL Developer 功能授权给特定的 Oracle 用户和角色。在一个开发数据库里，你将允许所有的开发者使用所有的功能；然而在产品数据库里，你将典型地对大多数用户禁止所有功能，以使他们不能改变数据库或者不能占用太多资源而影响运行。

通过授予 PL/SQL Developer 权限给角色，你可以为特定人群定制授权。你可以利用现有的映射到用户组的角色（例如 DBA 和 RESOURCE）或者你可以专门为 PL/SQL Developer 用户组创建一些角色。

22.1 允许授权

授权信息被储存在 sys 方案的 *plsqldev_authorization* 表里。只要数据库里没有这个表，所有用户就都有权使用 PL/SQL Developer 的所有功能。如果你连接到还没有授权的数据库并且选择来自于*文件*菜单的*授权*项目，你将得到下列问题：

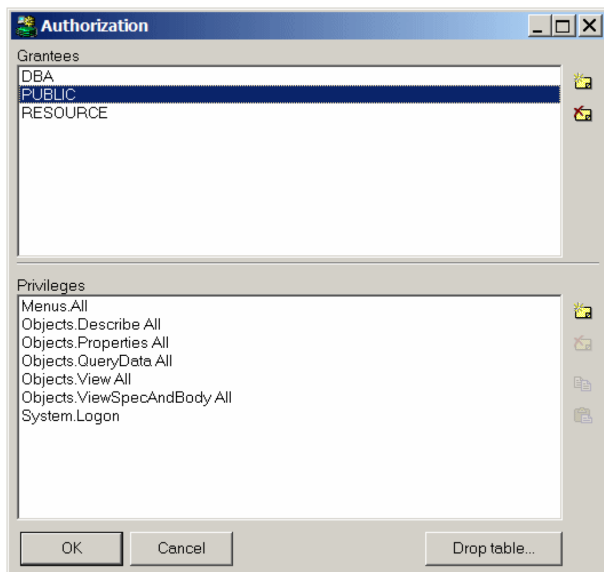


如果你选择*是*，授权就被允许了。除了 *SYS 之外*，PL/SQL Developer 就可以在当前的数据库实例上不再被所有用户使用了！注意，你必须被连接为 DBA 才能创建这个表。

在默认的情况下，只有表的所有者（sys）或另一个 DBA 才能改变表或更新它的内容。在表里，所有的用户都有 select 权限。

22.2 定义授权

要定义授权，你必须被连接为 DBA 并且启动来自于*文件*菜单的*授权*功能。下列对话框将显示出来：



在上部，你将看到所有的用户和角色，它们都具有一个或多个已经被授予的 PL/SQL Developer 的权限。在下部，你可以看到当前已选的被授予者的实际权限。

要添加一个被授予者，请按被授予者列表右边的*新建被授予者*按钮。这将调出一个所有潜在被授予者的列表（用户和角色）。你可以选择一个或多个（Ctrl-点击 或 Shift-点击）被授予者并按确定按钮来添加他们。在这里，这些新的被授予者还没有任何权限。

要移除一个被授予者和它的权限，请按*移除被授予者*按钮。这不会影响数据库里的实际用户和角色，只不过是从授权表里移除了它的权限而已。

要授予权限，请从列表里选择被授予者并按权限列表右边的*授予权限*按钮。一个 PL/SQL Developer 所有权限的列表就被显示出来了，在这里你可以选择一个或多个权限。共有三种权限：

- 系统权限 – PL/SQL Developer 系统功能（例如登录）。
- 对象权限 – 数据库对象类型特定权限（例如删掉表）。
- 菜单权限 – 使用 PL/SQL Developer 菜单功能的权限（例如，工具 > 会话）。

注意： 如果用户没有 *System.Logon* 或 *All* 权限，他或她就不能在当前数据库实例里使用 PL/SQL Developer 。

你可以选择单独的权限或者选择不同等级的权限。例如：

- All – 授予 PL/SQL Developer 的所有权限。
- Objects.All – 授予 PL/SQL Developer 的所有权限到数据库对象。
- Objects.Rename All – 授予 PL/SQL Developer 的所有的涉及到重新命名数据库对象的权限（重新命名表、视图、序列和同义词）。
- Objects.Rename Table – 授予 PL/SQL Developer 的用于重新命名表的权限。

注意，对于数据库对象，Oracle 用户或角色仍然必须有必要的系统权限以用于执行操作。授予 *Objects.Drop User* 权限仍然需要*删掉用户*的系统权限。

还要注意的是，如果用户有*删掉用户*的系统权限，他就能从 SQL 或 PL/SQL 脚本里执行*删掉用户*命令。

22.3 禁止授权

要永远禁止授权，你可以删掉 *sys.plsqldev_authorization* 表。要临时禁止授权，你既可以撤回在表里的权限又可以重新命名它（使它在 PL/SQL Developer 用户里不可见）。要在以后再一次允许它，你就再一次授予 select 权限。

23. 帮助系统

你可以配置 PL/SQL Developer 使之与 Oracle 服务器软件的帮助文件和手册结合起来。这一章描述了怎样设置和使用这些帮助系统。

23.1 MS 帮助文件

一直到 Oracle 7.2 ，在线手册都使用 MS 帮助格式。这些帮助文件可以被用于在编辑器里提供上下文有关的帮助，既可以通过右击编辑器里的一个词、通过选择*帮助*菜单里的 *SQL 帮助*或 *PL/SQL 帮助*项目来实现，又可以通过按你已经把这两个菜单项目关联起来的功能键来实现。

PL/SQL Developer 将在下列两个目录里搜索帮助文件：

1. 在 PL/SQL Developer 所在的目录里。
2. 在 SQL*Plus 所在的目录里。

SQL 帮助文件名被设为 *sqlhelp.hlp* ，PL/SQL 的帮助文件就是 *plshelp.hlp* 。如果这些帮助文件不可用，但你有一些想要使用的针对 SQL 或 PL/SQL 帮助的其它帮助文件，你就可以把这些文件放到 PL/SQL Developer 目录里并按前面提到的方法给它们重新命名。

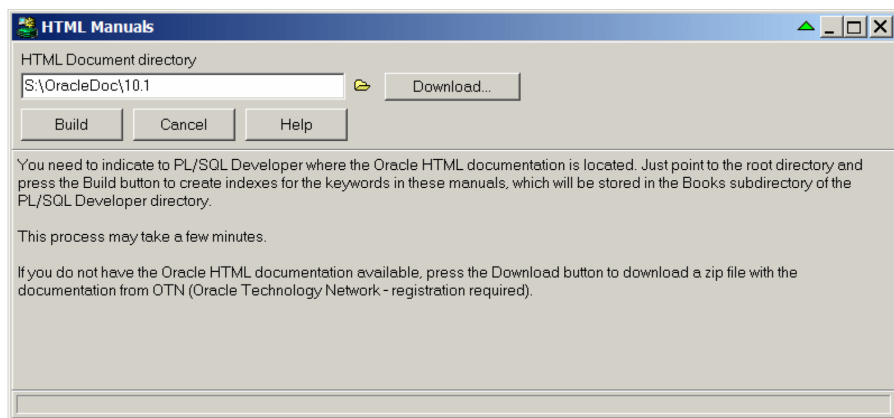
23.2 HTML 手册

自 Oracle 7.3 以来，在线手册已经被提供为 Acrobat Reader 和 HTML 格式。你可以通过配置 PL/SQL Developer 来使用 HTML 手册，既可以通过右击编辑器里的一个词、通过选择*帮助*菜单里的 *HTML 手册*项目来实现，又可以通过按 *F1* 来实现。此外，在 Oracle 出现异常的情况下，你可以在错误消息上按*帮助*按钮，这将自动地把你带到 *Oracle 错误信息*手册里的适当的段落。在程序窗口里双击一个编译错误也将显示出这个信息。

注意：一些版本的 Oracle 服务器的在线文档包含了超过 1MB 大小的 HTML 文件。载入这些文件能花费很长时间，在这样的情况下建议你使用章节被分成较小文件的一些版本的 Oracle 服务器的在线文档。对于 8.1.5 文档，Oracle 已经再一次减小了文件的大小。

配置 HTML 手册

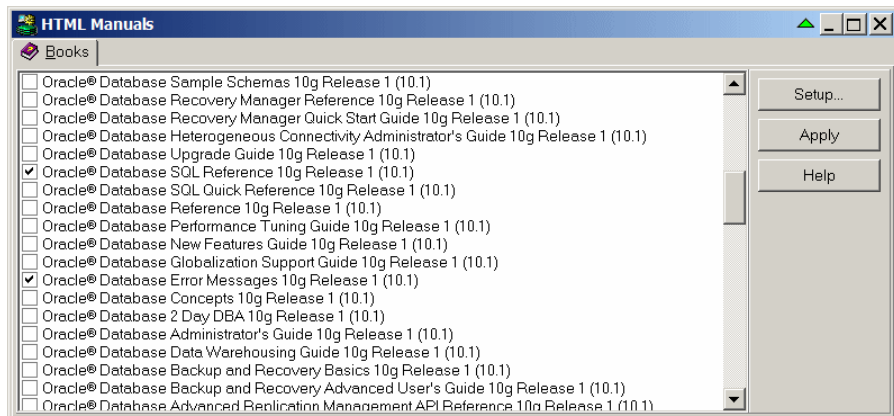
按照上面所描述的方法之一，你第一次调用 HTML 手册时，你需要告诉 PL/SQL Developer 你的 HTML 文件在什么地方：



仅仅指向根目录并且按*建立*按钮就可以在这些手册里为关键词创建索引了，这些索引被保存在 PL/SQL Developer 目录的 *Books* 子目录里。这个过程可能需要几分钟。

如果你没有可用的 Oracle HTML 文档，那么你就可以按*下载*按钮。这将把你带到本地的 HTML 页，在这里你可以选择适当版本的 Oracle 服务器并下载一个带有 HTML 文档的 zip 文件。这个下载需要你有一个 Oracle 技术网（OTN）的账号。如果你没有 OTN 账号，你可以先创建一个，它是免费的。

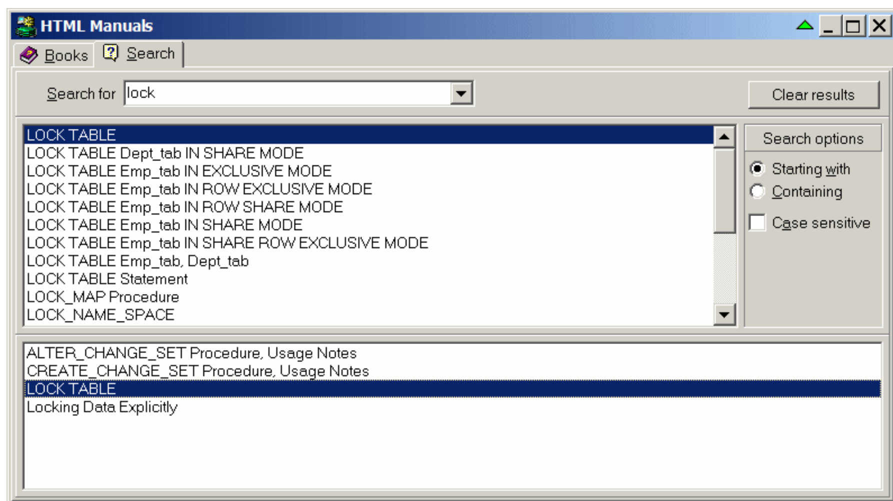
在建立索引完成之后，你会得到已经被找到的手册的列表。现在，你可以选择你要用的上下文关联帮助的书籍了。在默认的情况下，这些选择将包括*错误信息*、*SQL 参考*和 *PL/SQL 参考*手册，它们都包含了在 PL/SQL 开发期间的相关信息。在 Oracle 出现异常或编译错误的情况下，如果你要自动地得到有用的*原因*和*行为*，*错误信息*手册是必需的。当前的 Oracle7 手册不能适当地被检索以允许这个特性，Oracle8 版本就没有这个问题。



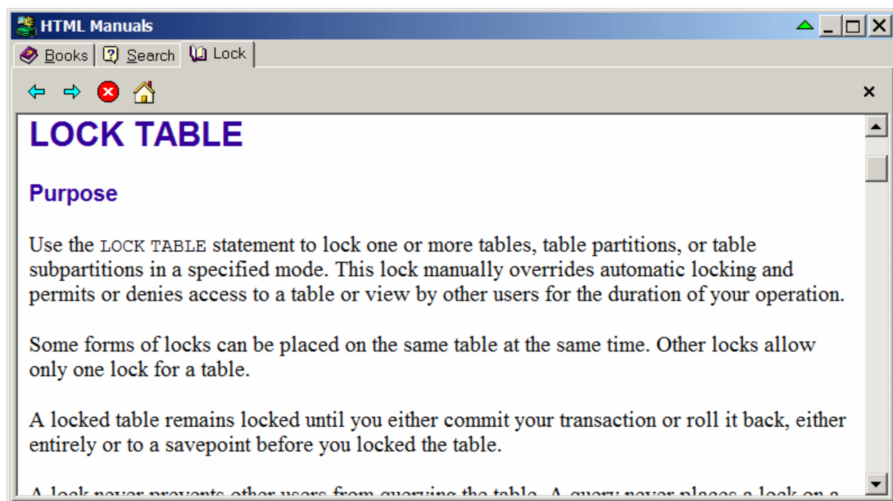
按*应用*按钮可使选择生效。在此之后，你会被带到搜索页，这将在下一章里讨论。通过按*设置*按钮，你可以为 HTML 手册扫描不同的目录。

使用 HTML 手册

在调用了 HTML 手册之后，在搜索页里，你可以键入你要搜索的关键词。有两个列表来显示匹配这个搜索和手册里的主题的关键词，在这里关键词出现了：



在关键词右边的*搜索选项*窗格里，你可以精制搜索规则。下一步你可以双击搜索页下部的主题列表里的项目来显示 HTML 手册里相应的段落：



搜索结果页显示了一个 HTML 页，在这个页里你通常可以通过超链接来操作。页顶部的按钮可以被用于操纵你以前要访问的上一个或下一个位置，还可以取消这个页组合，或者在操纵之后转到原始页。

你可以回到*搜索*页来搜索不同的关键词或者转到*书籍*页来选择不同的书籍。每一个搜索都创建了一个不同的标签页，所以你可以保存不同的有用的搜索结果。要关闭结果页，请按标签页右上方的关闭按钮。要关闭所有页，请按搜索页里的*清除结果*按钮。

在默认的情况下，HTML 窗口将停留在 PL/SQL Developer IDE 里的其它窗口的上方，所以信息的停留是可见的。要临时隐藏窗口，你可以使用窗口右上方的绿色的卷起按钮。

24. 定制

PL/SQL Developer 的一些方面可以被定制以满足你的个人需要。这些方面在下列段落里有描述。

24.1 首选项

在工具菜单里，有一个*首选项*项目允许你为 PL/SQL Developer 设置不同的首选项。这些首选项在第 14 章里有详细描述。

24.2 窗口版面

有多种窗口版面可以根据你的需要来制作，在*窗口*菜单里选择*保存版面*项目将保存这些版面。版面设置包括：

- 应用程序窗口的大小和位置。
- 对象浏览器的存在和大小。
- 程序窗口里代码目录的存在和大小。
- 窗口列表和模板窗口的存在、状态（浮动或停靠）和位置。
- 程序窗口、测试窗口、SQL 窗口、解释计划窗口、命令窗口、报告窗口、编译无效对象窗口、查找数据库对象窗口、导出用户对象窗口和比较用户对象窗口的大小。

当你保存版面时，最后打开的窗口的大小就被保存了。如果某一类型的窗口不存在，则那个窗口类型的大小设置就保持不变。

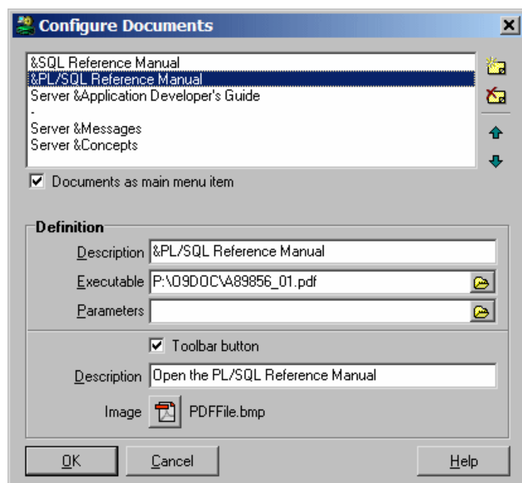
24.3 在线文档

大多数的 Oracle 文档都被提供为在线文档。随着时间的推移，这些文档已经有了几种格式。到目前为止，已经有了 MS 帮助文件、Oracle 书籍文件、Adobe PDF 文件和 HTML 文件。在将来，Oracle 也许还会选择使用另外一种格式。

此外，你可以有一些共同的标准和有时要访问的文字处理软件格式的工程文档。

通过在文档菜单里包括它们，所有这些在线文档都能被集成到 PL/SQL Developer 的 IDE 里。这样，通过一个简单的鼠标点击，它们对你就是可用的了。

要配置在线文档，请选择*工具*菜单里的*配置文档*项目。下列对话框允许你定义每一个文档的描述，以及怎样去查看它：



在右边，你可以看到四个按钮：在菜单里创建新文档、删除文档和上移、下移文档。当你创建或修改一个文档时，你必须提供下列信息：

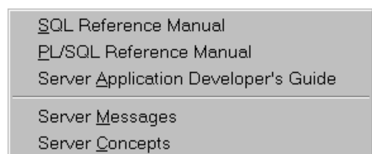
描述在菜单里显现出来。要为文档的菜单项目创建一个快捷键，就要在描述里的这个字母的前面加上一个 & 前缀。如果你要在描述里使用 & 字符，那么就用两个 & 字符以代之。如果你只输入一个“ - ”字符到描述里，那么它将变为菜单里的分隔线。这样，你可以把文档分到合理的组里。

可执行文件应该是你要用来查看文档的带有完整路径的程序。如果文档的类型是一个已被注册的文件类型（象 MS 帮助文件或 HTML 文件），你就可以在这里输入带路径的文档。

参数能被用于传递信息给可执行文件（如果它是程序的话）。它至少应该包括带路径的文档，但也能用于传递选项给程序。

当*作为主菜单项目的文档*选项被检验时，*文档*菜单就被定位在主菜单里。如果它没有被检验，它将被定位在*工具*菜单的下面。

在上面的例子里，PL/SQL Reference Manual 是一个 PDF 文件（P:\09DOC\A89856_01.pdf），它是一个已注册的文件类型，所以只能在可执行文件输入框输入文件名。字母 'P' 被用于菜单里的快捷方式。文档菜单将像这样：



当*工具栏按钮*选项被允许时，文档还能被包括到工具栏里。当你控制鼠标光标越过工具栏按钮时，相应的*描述*将被作为提示显示出来。如果你保留这个描述为空，则文档的主描述（与在菜单里显示的一样）将被使用。按*图象*按钮来为工具栏按钮选择一个 Windows 位图文件（*.bmp）。这个图象的大小应该是 20 x 20 像素比较合适。注意，PL/SQL Developer 将始终从原始位置载入位图文件，因此若相应的文档没有变动你就不要移除或重新命名这个文件。PL/SQL Developer 带有许多标准的位图文件，你可以从那里挑选它们。这些文件位于 PL/SQL Developer 安装目录里的 Icons 子目录里。这是图片选择器默认的目录。

24.4 命令行参数

下列参数可以被用于 PL/SQL Developer 命令行。默认的快捷方式不包括任何参数，只不过从 PL/SQL Developer 安装目录里启动 plsqldev.exe 而已。

通过 PL/SQL Developer 安装目录里的 *params.ini* 文件，默认参数可以被定义。你可以在像记事本这样的文本编辑器里编辑这个文件，这个文件包含了每一个参数的说明。如果参数在 *params.ini* 里和在命令行上被定义，那么命令行具有优先地位。

userid

每次你启动 PL/SQL Developer ，它将提示你至少要输入一个口令。要避免这个，你可以提供一个常见的用户名/口令@数据库格式的 userid 参数：

```
plsqldev.exe userid=scott/tiger@chicago
```

注意，你还可以为下列的注册表键提供一个默认的登录方法：

HKEY_CURRENT_USER\Software\Allround Automations\PL/SQL Developer\Logon

在这里，你可以添加 Username 、Password 和 Database 。

后一个方法的优点是，在双击了 PL/SQL Developer 已注册的文件之后，它能使你自动登录。

nologon

当 PL/SQL Developer 被启动时，Nologon 参数禁止了被显示的登录对话框。这个参数不需要任何自变量：

```
plsqldev.exe nologon
```

oraclehome

你可以在命令行上指定 Oracle 的主目录名，从而越过主 Oracle 主目录（被默认使用）和 Oracle 主目录首选项：

```
plsqldev.exe oraclehome=ora817
```

注意，这是 Oracle 主目录的<u>名字</u>，被指定在注册表里的 ORACLE_HOME_NAME 键里。这个名字也被 Oracle 主目录选择器使用，它<u>不</u>是注册表的组名或目录名。

dontadjustpath

PL/SQL Developer 将不为 Oracle Net 的初始化而临时修改 PATH ：

```
plsqldev.exe dontadjustpath
```

nosplash

当 PL/SQL Developer 被启动时，Nosplash 参数禁止了闪屏。这个参数不需要任何自变量：

```
plsqldev.exe nosplash
```

noplugins

指定 noplugins 参数将防止任何插件被载入：

```
plsqldev.exe noplugins
```

library

指定报告风格库的位置：

```
plsqldev.exe library=p:\standard.lib
```

prefpath

指定个人首选项集的位置。例如：

```
plsqldev.exe prefpath=u:\userdata
```

要了解详细资料，请看第 14.31 章。

commandfile

在命令窗口里运行指定的命令文件。例如：

```
plsqldev.exe userid=scott/tiger commandfile=”u:\sql scripts\demo_build.sql”
```

这个命令将首先连接到数据库然后运行 *demo_build.sql* 。

viewobject

查看指定数据库对象。它需要你也指定一个 *userid* 参数。例如：

```
plsqldev.exe userid=scott/tiger@chicago viewobject=emp
```

这个命令将首先连接到数据库，然后显示表 *EMP* 的表定义。注意，对象说明也可能包括所有者（例如 *SCOTT.EMP*）。

editobject

编辑指定数据库对象。它需要你也指定一个 *userid* 参数。例如：

```
plsqldev.exe userid=scott/tiger@chicago editobject=emp
```

这个命令将首先连接到数据库，然后显示表 *EMP* 的表定义。注意，对象说明也可能包括所有者（例如 *SCOTT.EMP*）。

24.5 SQL、PL/SQL、命令、Java 和 XML 关键词

在 SQL 、PL/SQL 和命令编辑器里高亮显示的关键词可以被定义在一个关键词文件里。在 PL/SQL Developer 的安装目录里，你可以找到文本文件 *sql.kwf* 、*plsql.kwf* 、*command.kwf* 、*java.kwf* 和 *xml.kwf* 。你可以使用像记事本这样的文本编辑器在适当的地方简单地改变、添加或移除关键词。

默认的关键词文件包含了 Oracle9i 的所有关键词。

24.6 插件

你可以规划你自己的插件以扩展 PL/SQL Developer 的功能。插件的功能可以添加到 PL/SQL Developer 的菜单里，它可以执行任何种类的任务：访问数据库、对象浏览器、当前窗口和编辑器等等。

插件是 DLL 文件，它有特定的接口与 PL/SQL Developer 连接。要创建一个插件，你可以使用任何能创建 DLL 文件的程序语言。当插件 DLL 文件被放到 PL/SQL Developer 的插件目录里时，它将自动被拾取。这使你分发你的插件变得很容易。

你可以建立你自己的或公用的插件。Allround Automations 网站上的一些标准的插件也是可以利用的（http://www.allroundautomations.com/plsqldev.html）。在这里，第三方插件也是可用的。

在 PL/SQL Developer 根目录下的 *PluginDoc* 子目录里，插件文件和例子也是可用的。